LINEAR PREDICTION AND POWER SPECTRUM ESTIMATION IN TWO-DIMENSIONS

A Thesis Submitted
In Partial Fulfilment of the Requirements
For the Degree of
MASTER OF TECHNOLOGY

By
M. N. PRAKASH

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JULY 1978

EE-1970-M-PRA-LIN

CERTIFICATE

This is to certify that the thesis entitled, "Linear Prediction and Power Spectrum Estimation in Two-Dimensions" is a record of the work carried out under my supervision and that it has not been submitted elsewhere for a degree.

Dr. S.K. Mullick
Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur

July 1978.

ACKNOWLEDGEMENT

I would like to record my sincere gratitude to Prof. S.K. Mullick for having suggested the problem and for his constant guidance. He devoted much of his valuable time for many sessions of discussions with me.

I would like to thank Mr. V.K. Joshi for his help in the preparation of the figures. I would also like to thank many other friends who helped me in the preparation of this report.

Thanks are due to Mr. K.N. Tewari for his neat and efficient typing of this thesis.

Kanpur
July 1978

M.N. Prakash

CONTENTS

ABSTRACT

In this report our main concern is linear prediction as well as spectral estimation in two dimensions. The various models applicable for two-dimensional prediction are given. Both conventional and AR-MLM model for two dimensional spectral estimation are discussed. The result includes many numerical examples, a detailed bibliography and a listing of related computer routines.

CHAPTER 1

INTRODUCTION

With the advent of digital computers and digital ICs, digital techniques for solving engineering problems have become increasingly popular. Using the digital approach, the designer no longer has to be concerned with realizability constraints of analog devices. Digital systems are concerned with discrete time inputs and their processing to achieve a desired objective. Before the highspeed digital computers were developed discrete time signal processing was limited to relatively simple algorithms. Advancements in digital computer technology revealed enormous potential of computers and digital signal processing has become a very important part of many areas of science and technology.

One of the most important areas of application, for digital signal processing techniques is the estimation of power spectrum of a time or space series. The need for power spectrum arises in a variety of contexts including the measurement of noise spectra for the design of optimal linear filters, the detection of narrow band signals in wide band noise, the estimation of parameters of a linear system by using a noise excitation etc.

The present work consists of two parts, namely: (1) two dimensional prediction, (2) two dimensional spectral analysis.

The spectral analysis of multi-dimensional signals plays a very important role in areas, such as, image processing, radio astronomical applications, geophysics and solid state physics. In radio astronomy itself it has many applications: aperture synthesis, interplanetary scintillation analysis and radar back scatter studies.

There are basically two conventional methods of power spectrum analysis, namely the autocorrelation or the Blackman-Tukey method and the "periodogram" method. Both these methods, which are nonparametric, make certain assumptions about the data outside the observation interval and their resolution performance is directly dependent on the length of the observation interval. Some of these limitations prompted the search for alternative schemes and one of the most promising one to come by in recent years is the so-called "maximum entropy method". The method is consistant with the statistical information available about the data and is "maximally non-committal" about the data outside the observation interval. The method has shown dramatic results in the resolution capability on short data lengths. Furthermore, the method yields results which are shown to be equivalent of fitting an optimum autoregressive or prediction model to the data under consideration.

The present report, therefore, begins with a discussion of autoregressive or prediction models for two

dimensional data. Chapter 2 deals with various types of predictive filtering. In each case the"normal equations" are derived using Minimum Mean Square Error Criterion (MMSE). Justice [18] has given an algorithm, (based on the theory of Szegö polynomials), which is analogous to the Levinson's algorithm in one dimension. This has been discussed in some detail. Chapter 3 deals with conventional methods of spectral estimation. Chapter 4 deals with the maximum entropy method in 2-dimensions. This chapter shows how Wiener filter theory can be extended to two-dimensional case and used for power spectral estimations. The concluding chapter presents and discusses some of the computational results. The computer programmes are listed in Appendix.

CHAPTER 2

LINEAR PREDICTION IN TWO DIMENSIONS

The mathematical analysis of general dynamic systems has been an area of major concern since the beginning of this century. With the advent of digital computers, this problem has been pursued with greatest vigour. The analysis of outputs of dynamic systems has been of fundamental importance in the fields of communications, statistics and economics. The 'time series' analysis as it is called is used in many areas of communications such as: speech and picture transmission, automatic speech recognition and data compression.

An important concern of time series analysis is the estimation of power spectra, cross spectra, and coherence functions. An important technique in many of these studies is 'Linear Prediction'. In linear prediction, we try to fit a linear model for the process under study, based on the statistics of the observed data. One such model is the Auto Regressive Moving Average (ARMA) model. In ARMA model, the signal S_n is considered to be the output of a system with some unknown input u_n, such that the following relation holds good.

$$S_n = - \sum_{k=1}^{p} a_k S_{n-k} + G \sum_{l=0}^{q} b_l u_{n-l}, \quad b_0 = 1 \qquad (2.0.1)$$

where a_k, $1 \leq k \leq p$, b_l, $1 \leq l \leq q$, and the gain G are the parameters of the hypothised system. Equation (2.0.1) says

that the output S_n at any instant of time is a linear combination of past outputs and present and past inputs. That is the signal S_n is predictable from the linear combination of past outputs and inputs. Hence the name 'Linear prediction'. Having fitted an ARMA model for the process, we can use the parameters of this ARMA model for different applications such as forecasting, data compression and power spectral estimation.

The philosophy of linear prediction can be extended to two or more dimensional data. Here instead of a signal S_n with one independent variable n, we have a signal $s_{k,l}$ with two spatial variables k and l. By making use of suitable strategies, it is possible to develop a set of equations which are similar to that in one dimensional case. These equations can be solved recursively to obtain the parameters of the model.

2.1 PREDICTION MODEL FOR SCANNED DATA:

In this and the following sections, we discuss various models suitable for two dimensional prediction. The one under consideration is suitable for scanned two dimensional data such as pictures [11]. Let us imagine that the given picture is scanned into N sequences which are stationary in nature (Fig.1). Let X(i,n) denote the nth sample of the ith sequence. Let the cross correlation between two sequences be defined as

$$R_{ij}(k) = R_{ji}(-k) = E[\, X(i,q)\, X(j,\, q+k)\,] \qquad (2.1.1)$$

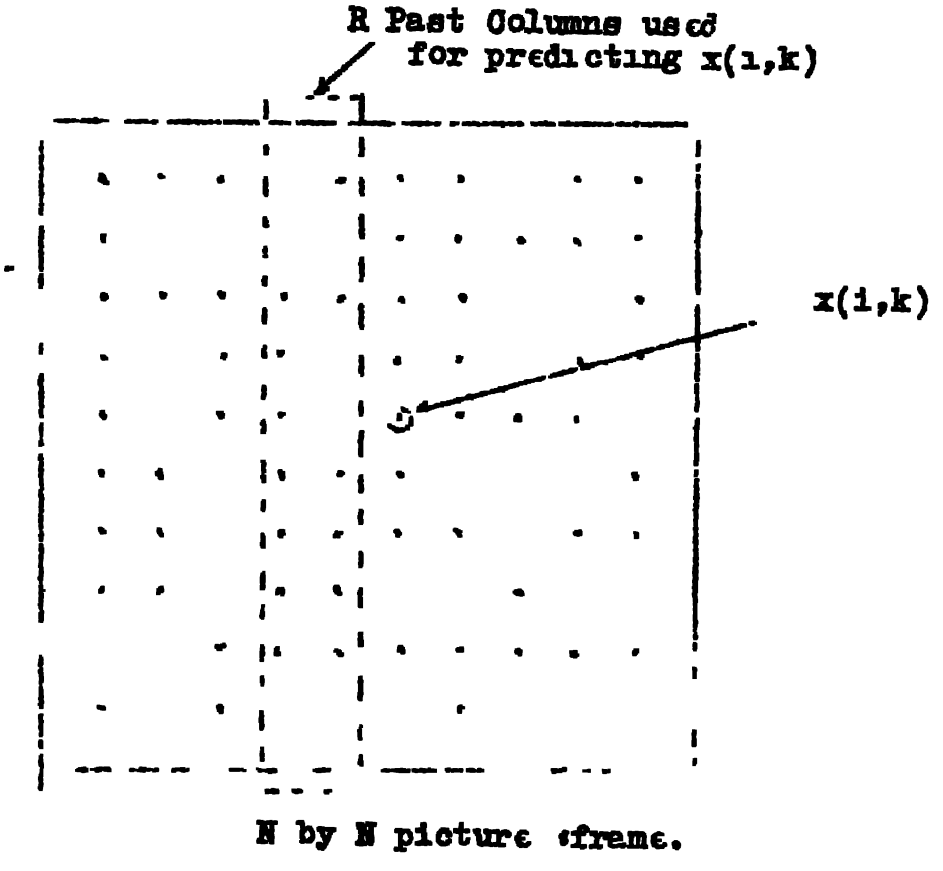

Fig.1 Schematic Arrangement for Prediction Strategy used in Section 2.1.

The prediction strategy is

$$\hat{X}(i,n) = \sum_{k=1}^{R} \sum_{j=1}^{N} \alpha\,(i,j,k)\; X(j,n-k) \qquad (2.1.2)$$

where $\alpha\,(i,j,k)$ are constants to be suitably chosen. Now we choose the coefficients such that the mean square error between the actual sample $X(i,n)$ and the predicted values $\hat{X}(i,n)$ is minimum.

Thus we are required to minimise the mean square error $\overline{\varepsilon^2}$ given by

$$\overline{\varepsilon^2} = E\,[\{X(i,n) - \hat{X}(i,n)\}^{2}]$$

$$= E\,[\{X(i,n) - \sum_{k=1}^{R} \sum_{j=1}^{N} \alpha(i,j,k)\; X(j,\, n-k)\}^{2}] \qquad (2.1.3)$$

Setting $\dfrac{\partial^2\, \overline{\varepsilon^2}}{\partial\alpha\;(i,j,k)} = 0$, we have

$$\frac{\partial^2\, \overline{\varepsilon^2}}{\partial\alpha\;(i,j,k)} = 2E\,[-X(j,\, n-k)\;\{X(i,n) - \sum_{q=1}^{R} \sum_{m=1}^{N} \alpha(i,m,q) X(m,n-q)\}]$$

$$= 2\,[-R_{ji}(k) + \sum_{q=1}^{R} \sum_{m=1}^{N} \alpha(i,m,q) R_{mj}(q-k)\,] = 0$$

Thus $\alpha(i,j,k)$'s are the solutions of the following equation.

$$R_{ji}(k) = \sum_{q=1}^{R} \sum_{m=1}^{N} \alpha(i,m,q)\; R_{mj}(q-k) \qquad (2.1.4)$$

Thus there are N^2R unknowns with N^2R equations and therefore these equations can be solved. This formulation represents the two dimensional generalisation of the prediction problem.

Let us try to put this in the matrix form. Toward this end, let

$$R_{ji}(k) = S(i,u)$$

$$\alpha(i,m,q) = \beta(i,v)$$

$$R_{mj}(q-k) = T(v,u)$$

where u is any unique one to one mapping of the ordered pairs (j,k) and v is the same mapping of the ordered pairs (m,q). For example, let

$$u = (j-1)R + k$$

$$v = (m-1)R + q$$

The prediction equation can then be written as

$$[S(i,u)] = [T(v, u)]\ [\beta(i,v)] \qquad (2.1.5)$$

where T(u,v) is a matrix with N^2 submatrices as shown below.

$$T(u,v) = \begin{bmatrix} [R_{11}(k-q)] & [R_{21}(k-q)] & \cdots & [R_{N1}(k-q)] \\ [R_{12}(k-q)] & [R_{22}(k-q)] & \cdots & [R_{N2}(k-q)] \\ \vdots & \vdots & \cdots & \vdots \\ [R_{1N}(k-q)] & [R_{2N}(k-q)] & \cdots & [R_{NN}(k-q)] \end{bmatrix}$$

Thus T(u,v) consists of submatrices which are Toeplitz matrices. Thus the equation (2.1.5) can be solved by using algorithms for inverting Toeplitz matrices [42].

2.2 WIENER FILTERING MODEL:

In this section, our main concern is to develop a two dimensional Wiener filter. The Wiener filter should be such that given an input array $\underline{b} = \{b_{kl}\}$ we must be able to obtain the desired output array $\underline{c} = \{c_{kl}\}$ as the convolution of the input array $\underline{b}$ and the filter array $\underline{a} = \{a_{kl}\}$. The formulation of the filter equations is as follows [18].

We are given arrays $\underline{b} = \{b_{kl}\}$ and $\underline{c} = \{c_{kl}\}$, $k \geq 0$, $l \geq 0$ satisfying

$$\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} |b_{kl}|^2 < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} |c_{kl}|^2 < \infty .$$

Our problem is to find a (N+1) by (M+1) array '$\underline{a}$' for which it is true that the convolution $b * a$ is the best least squares approximation to the array $\underline{c}$. Making use of the fact that the space of square summable sequences is a Hilbert space, we can derive a set of equation whose solutions yields the unknown array $\underline{a}$.

Define arrays u_{kl} as the translates of the $\underline{b}$ array i.e. $u_{kl}(i,j) = \underline{b}_{i-k,j-l}$ (zero for negative subscripts). We define an inner product by

$$\langle a,b \rangle = \Sigma\Sigma \; a_{kl} \; b^*_{kl}$$

and norm in the usual way,

$$\| a \|^2 = \langle a,a \rangle = \Sigma\Sigma \; |a_{kl}|^2$$

Our problem is then to minimise the norm

$$\| \sum_{j=0}^{N} \sum_{k=0}^{M} a_{kl}\, u_{kl} - c \|$$

That is, we must find the best approximation to $\underline{c}$ from the subspace spanned by the linearly independent arrays u_{kl}. It is well known that the solution satisfies the equation

$$< \sum_{k=0}^{N} \sum_{l=0}^{M} a_{kl}\, u_{kl} - c, \quad u_{st} > = 0 \quad \text{for } 0 \leq s \leq N, \quad 0 \leq t \leq M$$

That is, the difference array is orthogonal to the indicated subspace. We may rewrite the equations in the form

$$\sum_{k=0}^{N} \sum_{l=0}^{M} a_{kl}\ \phi_{s-k,t-l} = g_{st} \quad \text{for } 0 \leq s \leq N,\ 0 \leq t \leq M \tag{2.2.1}$$

These are "Normal equations".

The array $g_{st} = < c, u_{st} >$ is the "cross-correlation" of the desired array $\underline{c}$ with the input array $\underline{b}$. The array $\phi_{kl} = < b, u_{kl} >$ is the "auto-correlation" of the input array. It can be easily verified that a unique solution $\{ a_{kl} \}$ exists for any acceptable choice of the array $\underline{c}$. Also the definition of the coefficient $\{\phi_{kl}\}$ implies that

$$\phi_{kl} = \phi^{*}_{-k,-l}$$

Let us denote the row vectors of $\underline{a} = a_{jk}$ by $a_0, a_1, \ldots a_N$. Let $G_0, G_1, \ldots, G_N$ be the row vectors of the array $g = \{g_{rs}\}$.

The normal equation (2.2.1) then takes the form

$$\sum_{j=0}^{N} \Phi_{rj}\, a_j^T = G_r^T \qquad (2.2.2)$$

where Φ_{rj} is the (M+1) by (M+1) matrix given by

$$\Phi_{rj}(i,l) = \Phi_{r-j,i-l}$$

It is easy to see that $\Phi_{rj}(i,l) = \Phi_{r-j,i-l} = \Phi^*_{j-r,l-i} = \Phi^*_{jr}(l,i$

which is to say

$$\Phi_{rj} = \Phi^*_{jr}$$

Further, we see that $\Phi_{rj} = \Phi_{r+k,j+k}$.

If we define $\Phi_{k-l} = \Phi_{k,l}$, then the normal equations (2.2.2) assume the form

$$\begin{bmatrix} \Phi_0 & \Phi_1^* & \cdots & \Phi_N^* \\ \Phi_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Phi_1^* \\ \Phi_N & \cdots & \Phi_1 & \Phi_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_N \end{bmatrix} \qquad (2.2.3)$$

The matrix appearing in (2.2.3) is called "block Toeplitz" matrix, and we may easily verify that Φ_0 is a Hermitian matrix, in fact, positive definite and Φ_k, $k \neq 0$ is Toeplitz, but not Hermitian in general.

Our problem is to develop an efficient algorithm for solving the system of equations (2.2.3). Justice [18] has

analysed this problem in great detail and he has developed an algorithm based on the theory of Szego polynomials. He expands the unknown filter sequence, $\underline{a}$, in terms of the appropriate Stego polynomials, to obtain a recursive algorithm for calculating $\underline{a}$.

2.3 TWO DIMENSIONAL LEVINSON'S ALGORITHM [18] :

A. Calculation of the Bivariate Szego Polynomials:

The Block-Toeplitz matrix in (2.2.3) consists of blocks

$$\Phi = \{\Phi_{jk}\}, \qquad 0 \le j \le N, \quad 0 \le k \le N$$

The first row of Stego polynomials $p_{oo}, \ldots p_{om}$ is obtained by the following recursion relation:

a) $p_{o,o;o,o} = [\phi_{oo}]^{-\frac{1}{2}}$

b) $p_{o,l+1;o,j} = (1-|\lambda_l|^2 p^2_{o,l;o,l})^{-\frac{1}{2}} \cdot (p_{o,l;o,j-1} - \lambda_l p_{o,l;o,l} p^*_{o,l;o,l-j}) \quad 0 \le j \le l+1; \quad 0 \le l \le M-1$

where

$$\lambda_l = \sum_{k=o}^{l} \phi_{o,-(k+1)} \, p_{o,l;o,k}$$

Assuming now that the kth row of Szego polynomials has been calculated, we calculate the (k+1)th row as follows:

a) Calculate a polynomial $q_{k+1,l}$:

$$q_{k+1,l;i,j} = p_{k,l;i-1,j} - \sum_{t=o}^{l-1} \lambda_{kt} \, p_{k+1,t;i,j} - \sum_{t=o}^{M} \lambda'_{kt} \, p^*_{k,t;k-i,m-j} \qquad \begin{matrix} 0 \le i \le k+1, \\ 0 \le j \le M. \end{matrix}$$

where

$$\lambda_{kt} = \sum_{j=t}^{l} p^*_{k,l;k,j} \left(\sum_{r=0}^{k+1} \sum_{s=0}^{M} \phi_{k+1-r,j-s}\, p_{k+1,t;r,s} \right)$$

$$\lambda'_{kt} = \sum_{j=0}^{M} p_{k,t;k,m-j} \left(\sum_{r=0}^{k} \sum_{s=0}^{M} \phi_{(r+1),j-s}\, p_{k,l;r,s} \right)$$

Note that λ_{kt} and λ'_{kt} depend on the values of k and l. Any negative values in a subscript of a polynomial coefficient result in a zero value for the coefficient.

b) Finally, we obtain $p_{k+1,l}$ as follows:

$$p_{k+1,l;i,j} = q_{k+1,l;i,j} / d_{k+1,l}$$

where

$$d_{k+1,l} = \left(q^*_{k+1,l;k+1,l} \left(\sum_{r=0}^{k+1} \sum_{s=0}^{M} \phi_{k+1-r,l-s}\, q_{k+1,l;r,s} \right) \right)^{\frac{1}{2}}$$

This step simply normalises the corresponding Szego polynomial.

B. Levinson's Algorithm:

Having generated the orthogonal polynomials p_{kl}, $0 \le k \le N$, $0 \le l \le M$, as described in Section 2.3A, we calculate first iterate c_o to the desired filter as follows:

a) $$c_{o,j} = \sum_{k=0}^{N} \lambda_k\, p_{o,k;o,j}$$

where $$\lambda_k = \sum_{j=0}^{M} g_{o,j}\, p^*_{o,k;o,j}$$

and g_{rs} is the cross-correlation array of the input array and the desired output array.

b) Assuming kth iterate c_k has been calculated to calculate c_{k+1} by the equation

$$c_{k+1;i,j} = c_{k;i,j} + \sum_{l=0}^{M} \lambda_{kl}\, p_{k+1,l;i,j} \qquad \begin{matrix} 0 \le i \le K+1, \\ 0 \le j \le M \end{matrix}$$

where $$\lambda_{kj} = \sum_{r=0}^{k+1} \sum_{s=0}^{M} g_{rs}\, p_{k+1,j;r,s}$$

$$c_{0,j} = c_{0;0,j}$$

and $$c_{k;i,j} = 0 \quad \text{for } i > k.$$

The desired filter is obtained when c_N is obtained. As only the polynomials $p_{k+1,l}$ are needed to calculate c_{k+1} from c_k, the recursive calculation of the polynomials may be done simultaneously with the filter calculation, the previously calculated polynomials being discarded as the calculation proceeds.

2.4 GENERAL MODEL:

A model which is very important in spectral estimation is discussed here. This model is extensively used in spectral analysis by Maximum entropy method (MEM) discussed in Chapter 4. This is a causal model. Here the two-dimensional signal ξ_{ij} is thought to be composed of the message s_{ij} and noise n_{ij}. Thus we have

$$\xi_{ij} = s_{ij} + n_{ij} \tag{2.4.1}$$

where n_{ij} describes a discrete stochastic noise process.

On the other hand the signal is well defined, repeatable quantity. Hence we can extrapolate from the message's past history in order to predict with reasonable accuracy its future behaviour.

Thus we can estimate the message at (i,j) from a two dimensional neighbourhood of signals in any quadrant of the x-y plane. Thus

$$\hat{s}_{ij} = \sum_{k=0}^{Q} \sum_{l=0}^{R} g_{kl}\, \xi_{m-k,n-l} \; ; \quad g_{oo} = 0 \qquad (2.4.2)$$

Thus the noise filter for $n_{m,n}$ becomes

$$\hat{n}_{m,n} = \sum_{k=0}^{Q} \sum_{l=0}^{R} \Gamma^{*}_{kl}\, \xi_{m-k,n-l} \qquad (2.4.3)$$

where

$$\hat{n}_{m,n} = \xi_{m,n} - \overset{*}{s}_{m,n}$$

$$\Gamma_{k,l} = \begin{cases} 1 & k=l=o \\ -g^{*}_{k,l} & \text{otherwise} \end{cases}$$

Let $P = E[\, n_{m,n}\; n^{*}_{m,n}\,]$ be the noise power. Our aim is now to choose the coefficients Γ's such that noise power is minimised. The above formulation again leads to a set of equations which are block Toeplitz. The advantage of this formulation, called auto-regressive model (AR) lies in the fact that the model coefficients can be suitably deployed in determining the power spectral density.

The formulation of the equations and its extension to spectral analysis has been discussed in Chapter 4. The numerical results of the methods discussed in this chapter are given in Chapter 5.

CHAPTER 3

CONVENTIONAL METHODS OF POWER SPECTRUM ESTIMATION

Conventional methods for the estimation of power spectrum have been in use for quite a long time. In order to appreciate the advantages and the short-comings of some of the recent methods like the Maximum Entropy Method (MEM), one should have a clear understanding of the conventional methods. This chapter deals with the two popular techniques namely Blackman-Tukey (B-T) method and the "periodogram" approach for the power spectral estimation.

The first step that we encounter in the B-T method is the estimation of autocorrelation function. Thus in B-T method, in order to have a good estimate of the power spectrum we must know the autocorrelation values as accurately as possible.

The first section of this chapter deals with the estimation of autocorrelation values for two-dimensional arrays. The next section discusses the two conventional methods for power spectral estimation and their limitations. The last section deals with two-dimensional FFT and also two-dimensional windowing techniques.

3.1 ESTIMATION OF AUTOCORRELATION FUNCTION (ACF):

Power spectral density and ACF are related by the Fourier transform. In some cases even though the estimation

of power spectral density in the primary goal, the estimation of ACF is desired as an intermediate step.

Let us consider the estimation of ACF for two-dimensional data. Let X(i,j) be the two-dimensional array. Let $X'_{k,l}(i,j)$ denote the translated arrays, that is

$$X'_{k,l}(i,j) = X(i-k,\ j-l) \quad \text{(zero for negative subscripts)} \tag{3.1.1}$$

Then the autocorrelation function of the two-dimensional array is defined as

$$R_{xx}(k,l) = E\,[X(i,j)\ X(i-k,\ j-l)] \tag{3.1.2}$$

Our problem is then to estimate ACF from a finite observed data of size M by N. One such estimator is

$$C_{xx}(k,l) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j)\ X'_{k,l}(i,j)$$

$$= \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j)\ X(i-k,\ j-l) \tag{3.1.3}$$

$$\text{for all } |k| < M \text{ and } |l| < N$$

The immediate question that arises is the goodness of the estimator $C_{xx}(k,l)$. A good estimator should have an expected value that is close to the quantity being estimated. The difference between the true value of the quantity and the expected value of its estimator is called the <u>bias</u> of the estimator. The estimator is said to be <u>unbiased</u> if this bias is zero. From (3.1.3) the expected value of the

estimator is

$$E\,[C_{xx}(k,l)\,] = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} E\,[X(i,j)\,X(i-k,\,j-l)\,] \qquad (3.1.4)$$

$$= R_{xx}(k,l) \qquad \text{for } |k|<M\,,\ |l|<M$$

Thus we see that $C_{xx}(k,l)$ is an unbiased estimator. Another required property of a good estimator is that it should have a small variance. An estimator is said to be <u>consistent</u> if it converges in probability to the true value of the quantity being estimated as the size of the data becomes increasingly larger. A sufficient condition for an estimator to be consistent is that its bias and variance both converge to zero as M and N become infinite.

3.2 POWER SPECTRAL ESTIMATION:

Because of the limited observation interval, autocorrelation estimates increasingly deviate from the true values as k and l increase. Hence k and l are normally allowed to vary between the limits (-K, K) and (-L,L), where K is a small fraction of M and L is a small fraction of N. The Blackman-Tukey (B-T) method uses the previously mentioned estimate of ACF to obtain the power spectral density as

$$S(u,v) \triangleq \sum_{k=-K}^{K} \sum_{l=-L}^{L} C_{xx}(k,l)\, e^{-j(uk+vl)} \qquad (3.2.1)$$

where $j = \sqrt{-1}$ and u and v are the transform variables (e.g.

spatial frequencies in two-dimensional spatial data).

A modified spectral estimator

$$S'(u,v) \triangleq \sum_{k=-K}^{K} \sum_{l=-L}^{L} C_{xx}(k,l)\, h(k,l)\, e^{-j(uk+vl)} \qquad (3.2.2)$$

is more acceptable where h(k,l) is a two-dimensional 'lag window' which should be properly chosen. $S'(u,v)$ implicitly assumes that ACF is zero outside the lag (-K,K) and (-L,L). This zero extension is artificial and degrades the resolution performance of the spectral estimator. The two-dimensional windows h(k,l) are discussed in Section 3.3b.

Periodogram Technique:

This method employs the following estimator $I'_{MN}(u,v)$

$$I'_{MN}(u,v) = \frac{1}{MN} \left| \sum_{k=1}^{M} \sum_{l=1}^{N} X(k,l)\, e^{-j(uk+vl)} \right|^2 \qquad (3.2.3)$$

This method also turns out to be unacceptable, but various windowing techniques improve the estimator. The data window h(k,l) is chosen to improve stability and minimise leakage. This yields an improved spectral estimator $S''_{MN}(u,v)$.

$$S''_{MN}(u,v) = \frac{1}{MN} \left| \sum_{k=1}^{M} \sum_{l=1}^{N} X(k,l)\, h(k,l)\, e^{-j(uk+vl)} \right|^2 \qquad (3.24)$$

We observe that the term inside the modulus sign is the two-dimensional Fourier transform $F_{MN}(u,v)$ of the windowed data, i.e.

$$F_{MN}(u,v) = \sum_{k=1}^{M} \sum_{l=1}^{N} X(k,l)\, h(k,l)\, e^{-j(uk+vl)} \qquad (3.2.5)$$

Thus

$$S''_{MN}(u,v) = \frac{1}{MN} \left| F_{MN}(u,v) \right|^2 \qquad (3.2.6)$$

Digital computer implementation of the Fourier transform operation is best done using Fast Fourier Transform (FFT) techniques [7] . Because of inherent periodic nature of FFT, $S''_{MN}(u,v)$ implicitly assumes a periodic extension of the available data. This artificial extension again limits the resolution performance. The periodogram technique is favoured because it takes lesser computational time.

3.3 FAST FOURIER TRANSFORM (FFT) AND WINDOWING IN TWO DIMENSIONS:

a. <u>Two Dimensional FFT</u>:

FFT is an efficient algorithm for calculating the discrete Fourier transform (DFT) of given data. The DFT of a time series $X(n)$, $n = 0,1,\ldots,N-1$ is given by

$$X'(k) \triangleq \sum_{n=0}^{N-1} X(n)\, e^{-j\frac{2\pi}{N}kn} \qquad k = 0,1,\ldots,N-1 \qquad (3.3.1)$$

The inverse DFT is given by

$$X(n) \triangleq \frac{1}{N} \sum_{k=0}^{N-1} X'(k)\, e^{j\frac{2\pi}{N}kn} \qquad n = 0,1,\ldots,N-1 \qquad (3.3.2)$$

A direct approach for computation of the DFT using (3.3.1) requires $4N^2$ real multiplications and N(4N-2) real additions. Since the amount of computation, and thus the computation time is proportional to N^2, it is evident that the number of arithmetic operations required to compute the DFT by the direct method becomes very large for large values of N.

In 1965, Cooley and Tukey [43] published an algorithm for the computation of DFT that is applicable when N is a composite number. The publication of this paper created a flurry of activity in the application of DFT to signal processing and resulted in the discovery of a number of computational algorithms which have come to be known as Fast Fourier transform or simply FFT algorithms.

The fundamental principle that all these algorithms are based upon is that of decomposing the computation of the DFT of a sequence of length N into successive small discrete Fourier transforms. The manner in which this principle is implemented leads to a variety of different algorithms all with comparable improvement in computational speed. The principle of FFT can be easily extended to two-dimensions. The two-dimensional DFT of an array x(m,n) is defined as

$$X'(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n)\, e^{-j(w_1 km + w_2 ln)}$$

$$= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n)\, W_N^{ln}\, W_M^{km} \qquad \begin{matrix} k = 0,1,\ldots,M-1 \\ l = 0,1,\ldots,N-1 \end{matrix} \qquad (3.3.3)$$

where $W_N = e^{-j\frac{2\pi}{N}}$; $W_M = e^{-j\frac{2\pi}{M}}$

Writing (3.3.3) in a slightly different manner

$$X'(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(m,n)\, W_N^{ln}\, W_M^{km} \qquad (3.3.4)$$

We see that the computation of X'(k,l) involves M DFT's of the form

$$A(m,l) = \sum_{n=0}^{N-1} X(m,n)\, W_N^{ln} \qquad \begin{matrix} m = 0,1,\ldots,M-1 \\ l = 0,1,\ldots,N-1 \end{matrix} \qquad (3.3.5)$$

followed by N DFT's of the form

$$X'(k,l) = \sum_{m=0}^{M-1} A(m,l)\, W_M^{km} \qquad \begin{matrix} k = 0,1,\ldots,M-1 \\ l = 0,1,\ldots,N-1 \end{matrix} \qquad (3.3.6)$$

Thus the method for evaluating the two dimensional DFT using FFT algorithm is as follows.

1. Calculate the DFT of individual rows (or columns) of the given array X(k,l) of size (M by N) using N point (or M point) FFT.
2. Calculate the DFT of the individual columns (or rows) of the resultant array A(m,l) (or A(n,k)) using M point (or N point) FFT.

The subroutine FASTFT computes the two dimensional FFT of an array X of size N by N by the above method. This routine makes use of another routine FFT which evaluates the N point DFT.

b. Two-Dimensional Windowing:

Window functions are incorporated into the spectral estimator to improve the stability of the estimator. Excellent background material can be found in Blackman and Tukey [6], Openheim and Schafer [25] and Koopmans [19]. The window w(m,n) should be symmetric and also have unit amplitude at m = n = 0. Many useful windows have been designed. The one used in the present study is the Hamming window. Another The Appendix gives the routine HAMMIN for this window. Another important reason for using windows is to reduce leakage and improve resolution. Such a window should have minimum sidelobe and minimum main lobe width. Both these minima cannot be simultaneously achieved and hence a compromise must be made. Jenkins and Watts [16] point out that the important question in empirical spectral analysis is the choice of window bandwidth and not the choice of window function. The choice of windows in two-dimensions is much more complicated. Huang [13] has analysed the problem of windowing in two dimensions and has shown that by a suitable transformation, it is possible to get a good two dimensional window using a good one-dimensional window.

Let f(x,y) be a two-dimensional array. We would like to truncate this array using a window w(x,y). Let

$$g(x,y) = f(x,y)\, w(x,y) \qquad (3.3.1b)$$

where $w(x,y) = 0 \quad |x^2 + y^2| > A^2$.

The Fourier transform of g(x,y) is

$$G(u,v) = F(u,v) * W(u,v) \qquad (3.3.2b)$$

where F(u,v) is the Fourier transform of f(x,y) and W(u,v) is the Fourier transform of w(x,y). Given a good symmetrical one dimensional window w'(x), it is possible to find a two-dimensional window given by

$$w(x,y) = w'(\sqrt{x^2+y^2}) \qquad (3.3.3b)$$

For example, if

$$w'(x) = \begin{cases} 1 & |x| \leq 1 \\ 0 & |x| \leq 1 \end{cases} \qquad (3.3.4b)$$

W'(u) is given by

$$W'(u) = \frac{2 \sin u}{u} \qquad (3.3.5b)$$

This has a first side lobe whose peak is 0.2 times the peak at u = 0. The corresponding two-dimensional window using (3.3.3b) is

$$w(x,y) = \begin{cases} 1 & |x^2 + y^2| \leq 1 \\ 0 & |x^2 + y^2| > 1 \end{cases} \qquad (3.3.6b)$$

The two-dimensional Fourier transform has the form

$$W(u,v) = \frac{2\pi J_1(\sqrt{u^2+v^2})}{\sqrt{u^2+v^2}} \qquad (3.3.7b)$$

whose first side lobe is 0.12 times the peak at u = v = 0. A more general comparison would require a comparison of area under the side lobes in the case of 1-dimensional windows and

and volume under the side lobes in the case of two-dimensional windows. The following theorem gives such a comparison.

Theorem: If a symmetrical one-dimensional window $w'(x)$ and a two-dimensional window $w(x,y)$ are related by

$$w(x,y) = w'\left[(x^2 + y^2)^{\frac{1}{2}}\right]$$

then the Fourier transforms of $w'(x)$ and $w(x,y)$ satisfy the relation

$$\frac{1}{2\pi} W(u,v) * H(u,v) = W'(u) * H'(u)$$

where

$$H'(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases}$$

$$H(u,v) = \begin{cases} 1 & u \geq 0 \quad \text{for all } v \\ 0 & u < 0 \quad \text{for all } v \end{cases}$$

The proof of this theorem can be found in Huang [13].

The Windowing procedures are mostly aimed at smoothing the spectral estimate and this causes loss in resolution. This is a major problem in the analysis of short data records. Both B-T and the periodogram extend the data in an artificial manner without making use of any other available information the data provides. Burg [8] has suggested a method of spectrum estimation which makes use of the maximum entropy principle proposed by Jaynes [15].

This method known as Maximum Entropy Method (MEM) extends the data in a maximally noncommittal manner outside the given data interval. MEM results in a better resolution and is the subject of discussion in the next chapter.

CHAPTER 4

MAXIMUM ENTROPY METHOD (MEM) IN TWO-DIMENSIONS

As pointed out in the previous chapter, the conventional methods of power spectrum analysis make artificial assumptions about the data outside the given interval. Moreover, the window chosen for the purpose of improving the statistical properties of the estimate is prefixed and does not utilise the information available in the specific data being analysed. To overcome these drawbacks, Burg[8] proposed a new approach to spectral analysis which has come to be known as Maximum Entropy Method (MEM). The Burg's approach is based on the maximum entropy principle due to Jaynes [15].

The Burg's algorithm has been very profitably applied by Ulrych [31] to solve many problems in geomagnetism. The theoretical considerations concerning the development of MEM have been presented by Barnard [5] , Edward and Fitelson [12], and Smylie et al [29] Lacoss [20] has compared MEM with Maximum Likelihood Method. In an important communication, Van der Bos [36] has shown that the extrapolation of the ACF of a process by maximising the entropy of the corresponding probability distribution function, is in fact equivalent to least mean square fitting of auto-regressive (AR) model to the process. Using the duality

Akaike [3] has given a criterion for determining the length of the prediction error filter.

In this chapter, we discuss the formulation of MEM in two dimensions and show how the Wiener filter theory in two dimensions can be utilised for determining the prediction coefficients. The relationship between the prediction coefficients and the power spectrum of the signal is derived. The formulation of the"normal equations" and also an algorithm for solving these equations has been given.

4.1 TWO DIMENSIONAL POWER SPECTRUM:

For two dimensional signals, instead of the time variable,'t',we usually have two space variables x and y. In the frequency domain, instead of the single variable 'f', we have two spatial frequency variables and u and v. The corresponding power spectrum is denoted by S(u,v). We are interested in band limited processes only and hence we define A to be the set of all ordered pairs of spatial frequencies (u,v) such that $|u| \leq U_N$, and $|v| \leq V_N$. Then we impose

$$S(u,v) = 0 \qquad (u,v) \notin A \tag{4.1.1}$$

The power spectrum S(u,v) and the auto-correlation function $\rho(x,y)$ of the complex valued quantity $\xi(x,y)$ are related through

$$S(u,v) = \int_{-\infty}^{+\infty}\int \rho(x,y) \exp\{2\pi i(xu + yv)\}\,dxdy \tag{4.1.2}$$

$$\rho(x,y) = \int_{-\infty}^{+\infty}\int S(u,v) \exp\{-2\pi i(xu + yv)\}\,dudv$$

S(u,v) has the two dimensional Fourier series representation given by

$$S(u,v) = \Delta x \Delta y \sum_{\substack{m,n \\ -\infty}}^{+\infty} \rho_{m,n} \exp\{ 2\pi i(m\Delta x\, u + n\Delta y\, v)\} \qquad (4.1.3)$$

where $\rho_{mn} = \rho(m\,\Delta x,\ n\,\Delta y)$ are the sampled values of the autocorrelation function and

$$\Delta x = 1/2U_N \ ; \quad \Delta y = 1/2V_N \qquad (4.1.4)$$

$\rho_{m,n}$ is given by

$$\rho_{m,n} = E\{\xi_{k+m;l+n}\ \xi^*_{k,l}\} = \rho^*_{-m,-n} \qquad (4.1.5)$$

From (4.1.3) we see that a complete knowledge of the power spectrum needs a complete knowledge of the autocorrelation values. However, for finite length data, we can overcome this difficulty by making use of the Wiener filter theory and the maximum entropy principle.

4.2 POWER SPECTRUM FROM AUTO-REGRESSIVE MODEL (WIENER FILTER THEORY):

Let $\xi_{k,l}$ denote the received signal. This signal consists of a deterministic message $s_{k,l}$ and a random noise $n_{k,l}$. The noise is assumed to be zero mean white. Thus

$$\xi_{k,l} = s_{k,l} + n_{k,l} \qquad (4.2.1)$$

Since the message is deterministic and well defined, it normally possesses a Taylor series expansion in two dimensions.

Hence it is possible to estimate the signal at a point (m,n) from its neighbourhood of values in the (x-y) plane. Thus

$$\hat{s}_{m,n} = \sum_{k=o}^{Q} \sum_{l=o}^{R} g_{k,l}\, \xi_{m-k,n-l} \quad ; \quad g_{oo} = 0 \qquad (4.2.2)$$

For this signal filter, the noise filter becomes

$$\hat{n}_{m,n} = \xi_{m,n} - \hat{s}_{m,n} = \xi_{m,n} - \sum_{k=o}^{Q} \sum_{l=o}^{R} g_{k,l}\, \xi_{m-k,n-l}$$

i.e. $$\hat{n}_{m,n} = \sum_{k=o}^{Q} \sum_{l=o}^{R} \Gamma^{*}_{k,l}\, \xi_{m-k,n-l} \qquad (4.2.3)$$

where for $n = 0,\ldots,Q;\ l = 0,\ldots,R$

$$\Gamma_{k,l} = \begin{cases} 1 & k=l=0 \\ -g^{*}_{k,l} & \text{otherwise} \end{cases} \qquad (4.2.4)$$

Let P denote the average noise power,

$$P = E(n_{m,n}\, n^{*}_{m,n}) \qquad (4.2.5)$$

Then the noise power by employing (4.2.2) is

$$P_{Q+1,R+1} = E\left[\left| \sum_{k=o}^{Q} \sum_{l=o}^{R} \Gamma^{*}_{k,l}\, \xi_{m-k,n-l} \right|^{2} \right]$$

$$= \sum_{k,k'=o}^{Q} \sum_{l,l'=o}^{R} \Gamma^{*}_{k,l}\, \xi_{k',l'}\, \rho_{k-k',l-l'} \qquad (4.2.6)$$

Clearly the value of $P_{Q+1,R+1}$ depends upon the choice of $\Gamma_{k,l}$. If these are not properly chosen $P_{Q+1,R+1}$ contains a small faction of the message power also. Our aim is to choose $\Gamma_{k,l}$'s such that the contribution of message power

to the noise power is minimum. For this we take the derivative of $P_{Q+1,R+1}$ with respect to $\Gamma^*_{m,n}$ and set it zero for m=0,...Q, n = 0,...,R and m,n both not zero. Then we get

$$0 = \sum_{k=o}^{Q} \sum_{l=o}^{R} \Gamma_{k,l}\, \rho_{k-m,l-n} \qquad (4.2.7)$$

Introducing this into (4.2.6), we get

$$P_{Q+1,R+1} = \sum_{k=o}^{Q} \sum_{l=o}^{R} \Gamma_{k,l}\, \rho_{k,l} \qquad (4.2.8)$$

We now define coefficients $\gamma_{k,l}$ as

$$\gamma_{k,l} = \Gamma_{k,l} / (P_{Q+1,R+1})^{\frac{1}{2}} \qquad (4.2.9)$$

Equations (4.2.6) and (4.2.7) can now be written as

$$\sum_{k=o}^{Q} \sum_{l=o}^{R} \gamma_{k,l}\, \rho_{k-m,l-n} = \delta_{mo}\, \delta_{no}\, \frac{1}{\gamma^*_{oo}} \qquad (4.2.10)$$

$$m=o,\ldots,Q;\ n=o,\ldots,R.$$

In matrix form this becomes

$$\begin{bmatrix} \begin{bmatrix} \rho_{oo} & \cdots & \rho_{oR} \\ \vdots & & \vdots \\ \rho_{o,-R} & \cdots & \rho_{oo} \end{bmatrix} & \cdots & \begin{bmatrix} \rho_{Qo} & \cdots & \rho_{QR} \\ \vdots & & \vdots \\ \rho_{Q,-R} & \cdots & \rho_{Qo} \end{bmatrix} \\ \vdots & & \vdots \\ \begin{bmatrix} \rho_{-Q,o} & \cdots & \rho_{-Q,R} \\ \vdots & & \vdots \\ \rho_{-Q,-R} & \cdots & \rho_{Q,o} \end{bmatrix} & \cdots & \begin{bmatrix} \rho_{oo} & \cdots & \rho_{oR} \\ \vdots & & \vdots \\ \rho_{o,-R} & \cdots & \rho_{oo} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \gamma_{oo} \\ \vdots \\ \gamma_{oR} \\ \vdots \\ \gamma_{Qo} \\ \vdots \\ \gamma_{QR} \end{bmatrix} = \begin{bmatrix} 1/\gamma_{oo} \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (4.2.11)$$

An algorithm for determining γ_{kl}'s is given in Section 4.5. The γ's so determined can be used directly to determine the power spectrum S(u,v)as shown below.

4.3 POWER SPECTRUM S(u,v) IN TERMS OF γ'S:

The noise spectrum N(u,v) is defined as

$$N(u,v) = \begin{cases} \Delta x\,\Delta y \sum_{m,n=-\infty}^{+\infty} E[\, n_{k+m,l+n}\, n^{*}_{k,l}\,] \exp\{\, 2\pi i(m\,\Delta x\, u + n\,\Delta y v) & (u,v) \in A \\ 0 & (u,v) \notin A \end{cases} \tag{4.31}$$

For uniform white noise this becomes

$$N(u,v) = \begin{cases} \Delta x\,\Delta y\; P_{Q+1,R+1} & (u,v) \in A \\ 0 & (u,v) \notin A \end{cases} \tag{4.32}$$

Inserting (4.2.3) in (4.3.2) we have

$$N(u,v) = \left| \sum_{k=0}^{Q} \sum_{l=0}^{R} \Gamma^{*}_{k,l} \exp\{2\pi\, i(k\,\Delta x u + l\,\Delta y u)\} \right| \left| \sum_{k'=0}^{Q} \sum_{l'=0}^{R} \Gamma_{k',l'} \exp\{-2\pi\, i(k'\,\Delta x u + l'\,\Delta y v)\} \right| \left| \Delta x\,\Delta y \sum_{m,n=-\infty}^{+\infty} \rho_{m,n} \exp\{2\pi\, i(m\,\Delta x u + n\,\Delta y v)\} \right| \quad (u,v) \in A \tag{4.3.3}$$

From (4.1.3) we see that the last factor is nothing but S(u,v). Thus using (4.2.9), (4.3.2) and (4.3.3) we get the final result

$$S(u,v) = \frac{\Delta x\,\Delta y}{\left| \sum_{k=0}^{Q} \sum_{l=0}^{R} \gamma^{*}_{k,l} \exp\{-2\pi\, i(k\,\Delta x u + l\,\Delta y v)\} \right|^{2}_{(u,v) \in A}} \tag{4.3.4}$$

Thus by obtaining the values of γ_k from (4.2.11) and by using (4.3.4) the value of the signal power spectrum S(u,v) can be accurately determined.

4.4 THE MAXIMUM ENTROPY CONCEPT:

From information theory, the entropy H, of a discrete set, is given by

$$H = - \sum_{i=1}^{M} p_i \ln p_i \qquad (4.4.1)$$

where p_i is the probability of occurrence of an event m_i out of M possibilities. For a continuous process with probability distribution p(q), this becomes

$$H = - \int_{-\infty}^{+\infty} p(q) \ln p(q) \, dq \qquad (4.4.2)$$

For a Gaussian process

$$p(q) = \frac{1}{(2\pi e \sigma^2)^{\frac{1}{2}}} \exp\left[- \frac{(q-\mu)^2}{2\sigma^2}\right] \qquad (4.4.3)$$

where $E[q] = \mu$; $E[(q-\mu)^2] = \sigma^2$ (4.4.4)

The Entropy H then becomes

$$H = \ln(2\pi e \sigma^2)^{\frac{1}{2}} \qquad (4.4.5)$$

Having defined entropy let us see how the concept of maximum entropy fits into the determination of power spectral density. The problems that arise in conventional methods of power

spectral estimation are due to assumptions that are made concerning the data that lie outside the observation interval. These assumptions that constrain the data to be either periodic or zero outside the known interval are incorrect and are reflected in performance degradation of the spectral estimator. What is required is then an approach that is first of all, consistent with the prior knowledge and second, estimates the prior probability assignment that describes the prior information without assuming anything beyond that. The relationship between entropy and uncertainty led Jaynes [15] to the formulation of maximum entropy principle which can be stated in the following manner:

"The prior probability assignment that describes the available information but is maximally non-committal with regard to the unavailable information is the one with the maximum entropy".

In order to apply the concept of maximum entropy to spectral analysis, we begin with the relationship between the entropy and the spectral density of a Gaussian process. In all but most pathological cases signals possess a Fourier representation. We associate to each Fourier oscillator $\exp(-2\pi\nu t)$, at frequency ν, a comlex valued Fourier oscillator strength $f(\nu)$. For a stochastic process $f(\nu)$ is a random quantity having some probability distribution function. It is convenient to employ a Gaussian distribution in amplitude

and a uniform distribution in phase to independently describe the oscillator strengths at each frequency. Thus

$$E[f(v)] = 0 \; ; \quad S(v) = E[f(v)\, f^*(v)] \tag{4.4.6}$$

where S(v) is the power spectrum. From (4.4.5), we have

$$H(v) = \ln [2\pi e\, S(v)]^{\frac{1}{2}} \tag{4.4.7}$$

We define the average entropy over the bandwidth to be

$$H_B = \frac{1}{2V_N} \int_{-V_N}^{+V_N} H(v)\, dv = \frac{1}{4V_N} \int_{-V_N}^{+V_N} \ln [S(v)]\, dv + \ln(2\pi e)^{\frac{1}{2}} \tag{4.4.8}$$

Given autocorrelation function values $\rho_{-M}, \rho_{-M+1}, \ldots, \rho_M$ the maximum entropy principle may be stated as follows: Maximise H_B with known value of autocorrelation function values

$$\rho_n = \int_{-V_N}^{+V_N} S(v) \exp\{-2\pi i\, n \Delta t\, v\}\, dv \; ; \; n = -M, \ldots, M \tag{4.4.9}$$

The solution to this problem is called the maximum entropy power spectrum. Since $\rho_{-M}, \ldots, \rho_M$ are known we vary ρ_n, $|n| > M$ so that H_B is maximised, that is

$$\frac{\partial H_B}{\partial \rho_n} = 0 \; , \quad |n| > M \tag{4.4.10}$$

From (4.4.8) we have

$$\int_{-V_N}^{+V_N} \frac{1}{S(v)} \frac{\partial [S(v)]}{\partial \rho_n}\, dv = 0 \quad \text{for } |n| > M \tag{4.4.11}$$

For a band limited spectrum between $(-V_N$ and $V_N)$ we have

$$S(v) = \begin{cases} \Delta t \sum_{n=-\infty}^{+\infty} \rho_n \exp(2\pi i\, n\, \Delta t\, v) & |v| \le V_N \\ 0 & |v| > V_N \end{cases} \qquad (4.4.12)$$

Thus H_B is an extremum when

$$\int_{-V_N}^{+V_N} \frac{1}{S(v)} \exp(2\pi i\, n\, \Delta t\, v)\, dv = 0 \qquad |n| > M \qquad (4.4.13)$$

Thus the Fourier series for $1/S(v)$ is truncated and has the form

$$S(v) = \left[\sum_{n=-M}^{M} a_m \exp(-2\pi i\, n\, \Delta t\, v) \right]^{-1} \qquad (4.4.14)$$

Since $S(v)$ is a real quantity, coefficients a_m are Hermitian, that is,

$$a_{-m} = a_m^* \qquad m = 0,\ldots,M \qquad (4.4.15)$$

By expanding (4.3.4) and comparing with (4.4.14) we get

$$a_m = 2V_N \sum_{l=\max(0,-m)}^{\min(n-m,m)} \gamma_{l+m}\, \gamma_l^* \qquad m = -M,\ldots,M \qquad (4.4.16)$$

Maximum Entropy in Two-Dimensions:

We can extend the maximum entropy principle to two dimensions. In two dimensions H_B is given by

$$H_B = \frac{1}{8U_N V_N} \int_{-U_N}^{U_N} \int_{-V_N}^{V_N} \ln[S(u,v)]\, du\, dv + \ln(2\pi e)^{\frac{1}{2}} \qquad (4.4.17)$$

where

$$S(u,v) = \Delta x \Delta y \sum_{\substack{-\infty \\ m,n}}^{+\infty} \rho_{m,n} \exp\{2\pi i(m\, \Delta x\, u + n\, \Delta y\, v)\} \qquad (4.4.18)$$

and

$$\rho_{m,n} = \int_{-U_N}^{U_N} \int_{-V_N}^{V_N} S(u,v) \exp\{-2\pi i(m\,\Delta x\,u + n\,\Delta y\,v)\}\,du\,dv \tag{4.4.19}$$

Maximum entropy principle then says that the unknown $\rho_{m,n}$ should be varied such that H_B is maximised. That is,

$$\frac{\partial H_B}{\partial \rho_{m,n}} = 0 \qquad |m| > Q \text{ or } |n| > R \tag{4.4.20}$$

Differentiating (4.4.17) we see that the average entropy is an extremum when

$$\int_{-U_N}^{U_N} \int_{-V_N}^{V_N} \frac{\exp\{2\pi i(m\,\Delta x\,u + n\,\Delta y\,v)\}}{S(u,v)}\,du\,dv = 0 \qquad |m| > Q \text{ or } |n| > R \tag{4.4.21}$$

Thus the Fourier transform of S(u,v) is truncated and has the form

$$S(u,v) = \left[\sum_{m=-Q}^{Q} \sum_{n=-R}^{R} a_{m,n} \exp\{-2\pi i(m\,\Delta x\,u + n\,\Delta y\,v)\}\right]^{-1} \tag{4.4.22}$$

For S(u,v) to be real the coefficients $a_{m,n}$ are Hermitian. That is,

$$a_{m,n} = -a^*_{m,n} \qquad m = -Q,\ldots,Q;\quad n = -R,\ldots,R \tag{4.4.23}$$

The coefficients $a_{k,l}$ and $\gamma_{k,l}$ are related by the equation

$$a_{m,n} = 4U_N V_N \sum_{k=\max(0,-m)}^{\min(Q-m,\,Q)} \sum_{l=\max(0,-n)}^{\min(R-n,\,R)} \gamma_{k+m,l+n}\,\gamma^*_{k,l} \tag{4.4.24}$$

4.5 ALGORITHM FOR CALCULATING TWO DIMENSIONAL FILTER COEFFICIENTS:

The following algorithm can be employed to solve the equation (4.2.11). This algorithm is due to Newmann[24] which is an adoption of Rissanen's method for solving block equidiagonal matrices. Let r_n be the submatrix given by

$$r_n = \begin{vmatrix} \rho_{no} & \cdots & \rho_{nR} \\ \vdots & & \vdots \\ \rho_{n,-R} & \cdots & \rho_{no} \end{vmatrix} \qquad n = -Q,\ldots,+Q \qquad (4.5.1)$$

r_n is Hermitian. That is

$$r_n^H = [r_n^*]^T = r_{-n} \qquad (4.5.2)$$

Let $\underline{G}_k$ and $\underline{\Delta}_k$ be two column vectors of length R+1.

$$\underline{G}_k = [\gamma_{ko}, \ldots, \gamma_{kR}]^T \qquad k = 0,\ldots,Q \qquad (4.5.3)$$

$$\underline{\Delta}_k = [\delta_{ko}, 0, \ldots, 0]^T$$

The equation (4.2.11) then becomes

$$\underline{\Delta}_m = \sum_{k=o}^{Q} \gamma_{k-m}\, \underline{G}_k \; ; \qquad m=0,\ldots,Q \qquad (4.5.4)$$

Let U_o^o and D_o^o be two matrices of size (R+1) by (R+1) given by

$$U_o^o = [r_o]^{-1} r_1 \; ; \quad D_o^o = [r_o]^{-1} r_{-1} \qquad (4.5.5)$$

Now iterate from N = 0, ..., Q-2

$$C_o^{N+1} = \left[\sum_{n=o}^{N+1} r_{N+1-n} D_n^N - r_o \right]^{-1} \left[\sum_{n=o}^{N} r_{n+1} C_n^N - r_{N+2} \right]$$

$$D_o^{N+1} = \left[\sum_{n=o}^{N+1} r_{n-N-1} C_n^N - r_o \right]^{-1} \left[\sum_{n=o}^{N} r_{-n-1} D_n^N - r_{-N-2} \right] \qquad (4.5.6)$$

With for n = 0,...,N

$$C_{n+1}^{N+1} = C_n^N - D_{N-n}^N C_o^{N+1}$$

$$D_{n+1}^{N+1} = D_n^N - C_{N-1}^N D_o^{N+1} \qquad (4.5.7)$$

C and D matrices satisfy

$$\sum_{n=o}^{N} r_{n-k} C_n^N = r_{n+1-k}$$

$$\sum_{n=o}^{N} r_{k-n} D_n^N = r_{k-n-1} \qquad k = 0,1,\ldots,N \qquad (4.5.8)$$

We define (R+1) by (R+1) matrix P by

$$P = \left[r_o - \sum_{n=o}^{Q-1} r_{Q-n} D_n^{Q-1} \right]^{-1} \qquad (4.5.9)$$

and D_Q^{Q-1} by

$$D_Q^{Q-1} = -I$$

where I is (R+1) by (R+1) identity matrix. Then we have

for k = 0,...,Q

$$\sum_{n=0}^{Q} r_{n-k} D_{Q-n}^{Q-1} P = -\delta_{ko} I$$

and

$$\sum_{n=0}^{Q} r_{n-k} D_{Q-n}^{Q-1} P \underline{A}_o = -\underline{A}_k \qquad (4.5.10)$$

The second equation is identical to (4.5.4) if we make the association

$$\underline{G}_k = -D_{Q-k}^{Q-1} P \underline{A}_o \qquad k = 0,\ldots,Q \qquad (4.5.11)$$

This algorithm requires $O(Q^2R^3)$ operations; to solve (4.2.11). Otherwise we need $O(Q^3 R^3)$ operations.

The routine MEM gives the Fortran IV version of the above algorithm. The next chapter contains many numerical examples and discusses the merits and demerits of the various methods of prediction and spectral estimation discussed in this and the previous chapters.

CHAPTER 5

RESULTS AND DISCUSSIONS

This chapter presents some of the computational results obtained for the prediction and power spectral estimation methods outlined in the previous chapters. Section 1 gives the numerical results for AR-MEM model for prediction. Section 2 discusses the AR-MEM model fitting. Section 3 gives the numerical results for the MEM, as well as B-T method of spectral estimation. The numerical computations were carried out on IBM 7044 and plots were drawn on IBM 1800.

5.1 AR-MEM MODEL OF PREDICTION (SEC.2.4):

The data used for this prediction is the binary data of a "girl's picture". The size of the data is 16 by 16. For this data the prediction coefficients are determined using AR-MEM algorithm. The picture is then reconstructed using these coefficients and the neighbouring points starting with known boundary values. Fig.2 gives the original picture as well as the predicted picture. The two levels used in this picture are 2.0 for 'blank' and 6.0 for 'star'. A better result may be expected for a picture with more number of gray levels, as the number of sharp edges in that case would be limited. The model order used in the present case is R = Q = 2.

THE ORIGINAL PICTURE

THE PREDICTED PICTURE

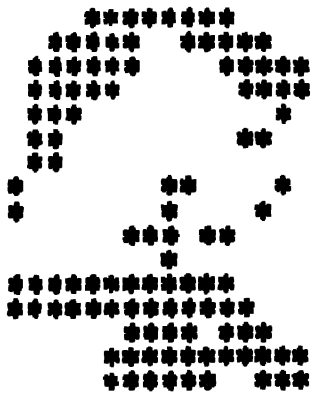

Fig 2

5.2 MODEL FITTING:

To test the efficacy and numerical accuracy of the AR-MEM model the scheme as depicted in Fig.3 was adopted. Here a two dimensional white noise is passed through a first order difference equation model, with known coefficients G(k,l), to obtain the two dimensional data x(k,l). Using the known coefficients G(k,l) the 'true spectrum' S(u,v) is determined. The two-dimensional data x(k,l) is then passed through the AR-MEM model to obtain the estimated parameters $\hat{G}(k,l)$. Using these estimates the 'MEM spectrum' $\hat{S}(u,v)$ is obtained. The two spectrums are compared. The Table 5.1 gives the actual and the estimated parameters for various difference equation models of order 1. Fig.4 gives the 'true spectrum' S(u,v) and Fig.5 gives the MEM spectrum $\hat{S}(u,v)$ for the first model (Table 5.1), Sl.No.1).

5.3 POWER SPECTRUM RESULTS:

To test the accuracy of the model, a two-dimensional sinewave of amplitude 10.0 and centered at frequency $(2\pi/6, 2\pi/6)$ was added to white noise of variance 1.0 and mean zero. This data was analysed using the above model. Fig.6 gives the spectrum of this data for model order 1. Fig.7 gives the spectrum of the same data for model order 2. Thus we see that increasing the model order gives better results. However, there is an optimum value of the model

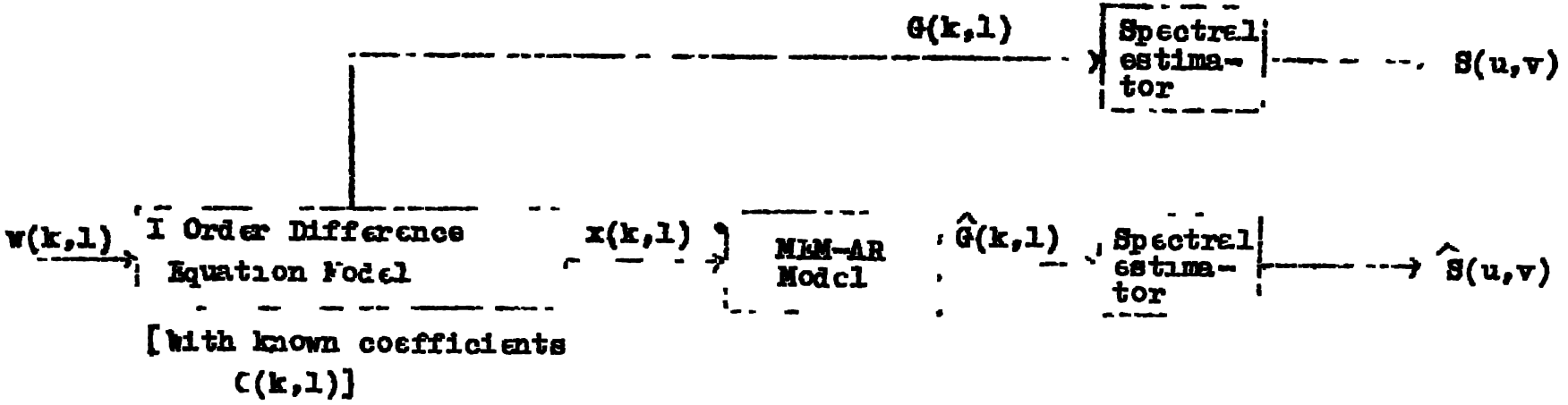

Fig.3 Schematic Arrangement for Model Testing.

Table 5.1

Sl.No.	Coefficient G(k,l)	Actual	AR-MEM	% Error
1	G_{oo}	0.95989	0.92401	3.68
	G_{ol}	-0.33596	-0.32321	3.78
	G_{1o}	-0.33596	-0.38784	14.15
	G_{11}	-0.14398	-0.13195	8.33
2	G_{oo}	0.93515	0.87944	5.9
	G_{o1}	-0.37406	-0.36653	2.2
	G_{1o}	-0.37406	-0.32715	10.9
	G_{11}	-0.16833	-0.12994	19.8
3	G_{oo}	0.92420	0.86025	7.1
	G_{o1}	-0.30499	-0.30499	0.0
	G_{1o}	-0.30499	-0.26920	10.6
	G_{11}	-0.30499	-0.30562	0.16
4	G_{oo}	0.91954	0.84909	7 5
	G_{o1}	-0.59770	-0.53025	11.4
	G_{1o}	-0.59770	-0.51517	13.6
	G_{11}	0.36782	0.31170	15.1

order to be used. Fig.8 gives the same sinewave spectrum for B-T method.

Resolution Performance:

To test the resolution performance of B-T as well as MEM method, a composite sinewave was used. The amplitude of the sinewave 10.0, The two frequencies were ($2\pi/12$, $2\pi/12$) and ($2\pi/6$, $2\pi/6$). The composite sinewave was sampled to obtain a 24 by 24 data. This was then mixed with white noise of variance and mean zero. The AR-MEM algorithm was then used to obtain the spectrum of this data. It was observed that for lower order models, MEM failed to resolve the two ponks (Fig.9, model order 1). As the model order was increased, the two peaks became more clear (Fig.10, model order 2). For model order 5, the two peaks were clearly resolved (Fig.11). However, spectral splitting was observed in one of the peaks. This spectral splitting has been observed by others also [44]. Fig.12 gives the same composite sinewave spectrum for B-T method.

[FOR FIGS 4, 5, 6, 7, 9, 10, 11, 13]
SCALE-X : 24 GRIDS
SCALE-Y : 24 GRIDS
[FOR FIG 8, 12]
SCALE-X : 32 GRIDS
SCALE-Y : 32 GRIDS

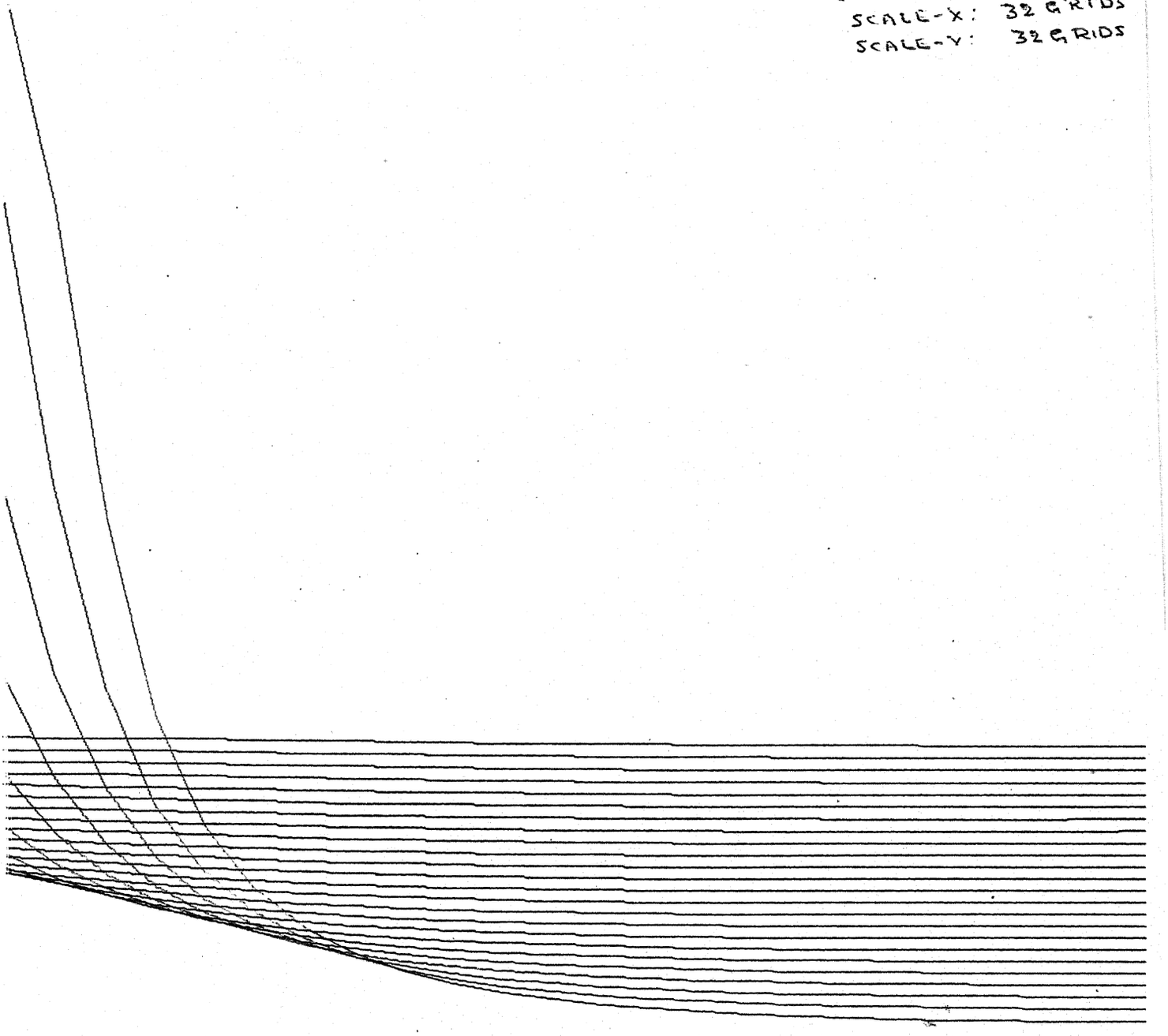

Fig.4: True Spectrum of the signal x(k,l) generated by the first order difference equation model (using the true coefficients G(k,l)).

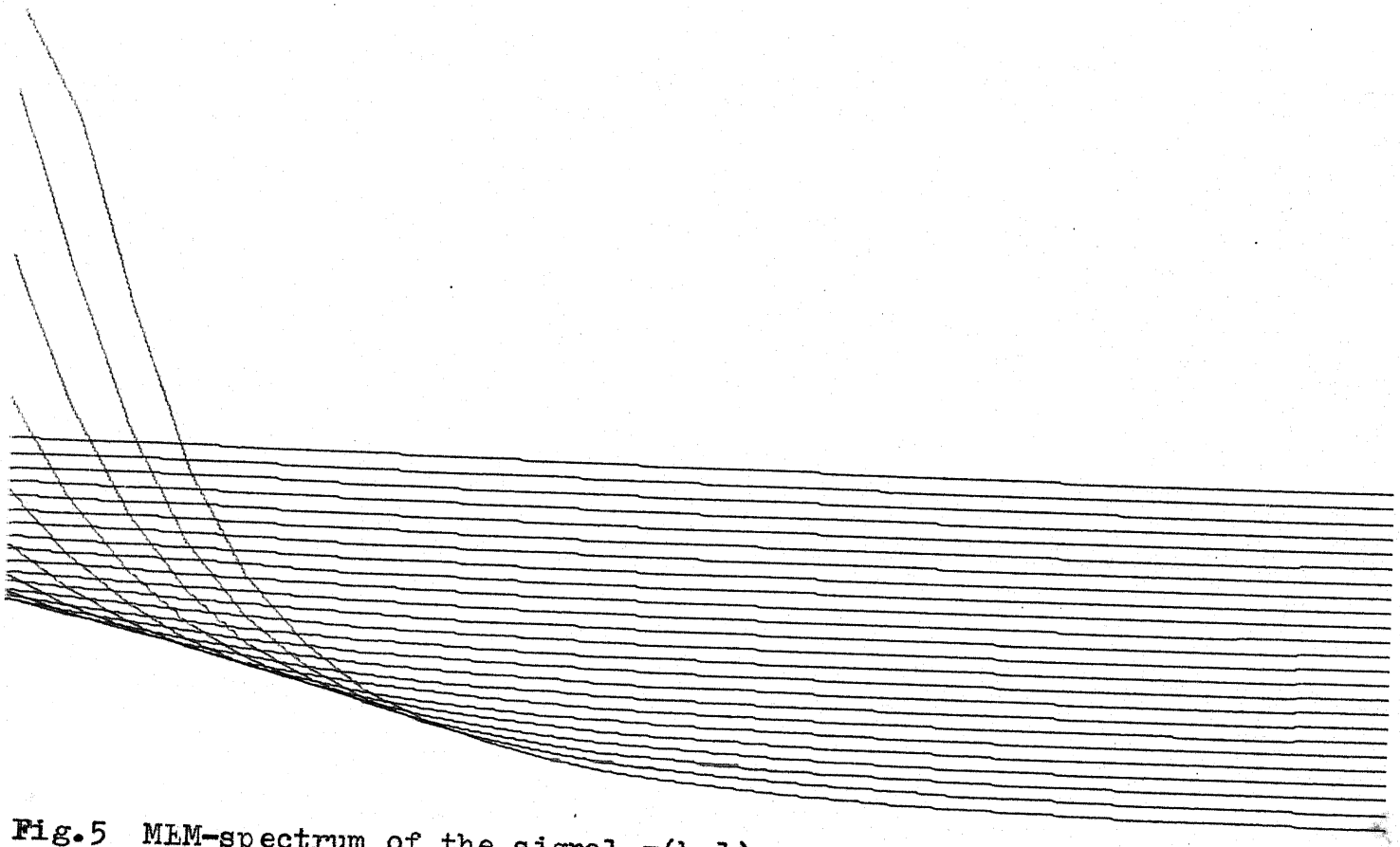

Fig.5 MLM-spectrum of the signal x(k,l) generated by the first order difference equation model [using estimated coefficients G(k,l)].

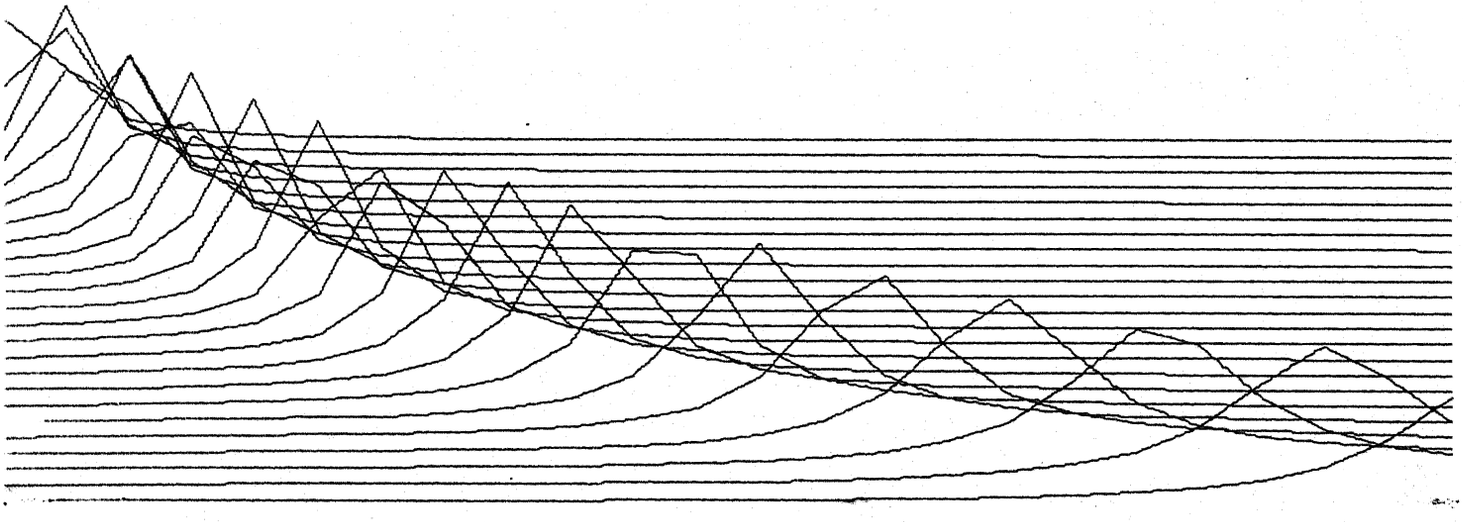

Fig. 6 Spectrum of sine wave centered at frequencies [2π/6, 2π/6] using AR-MEM model of order 1.

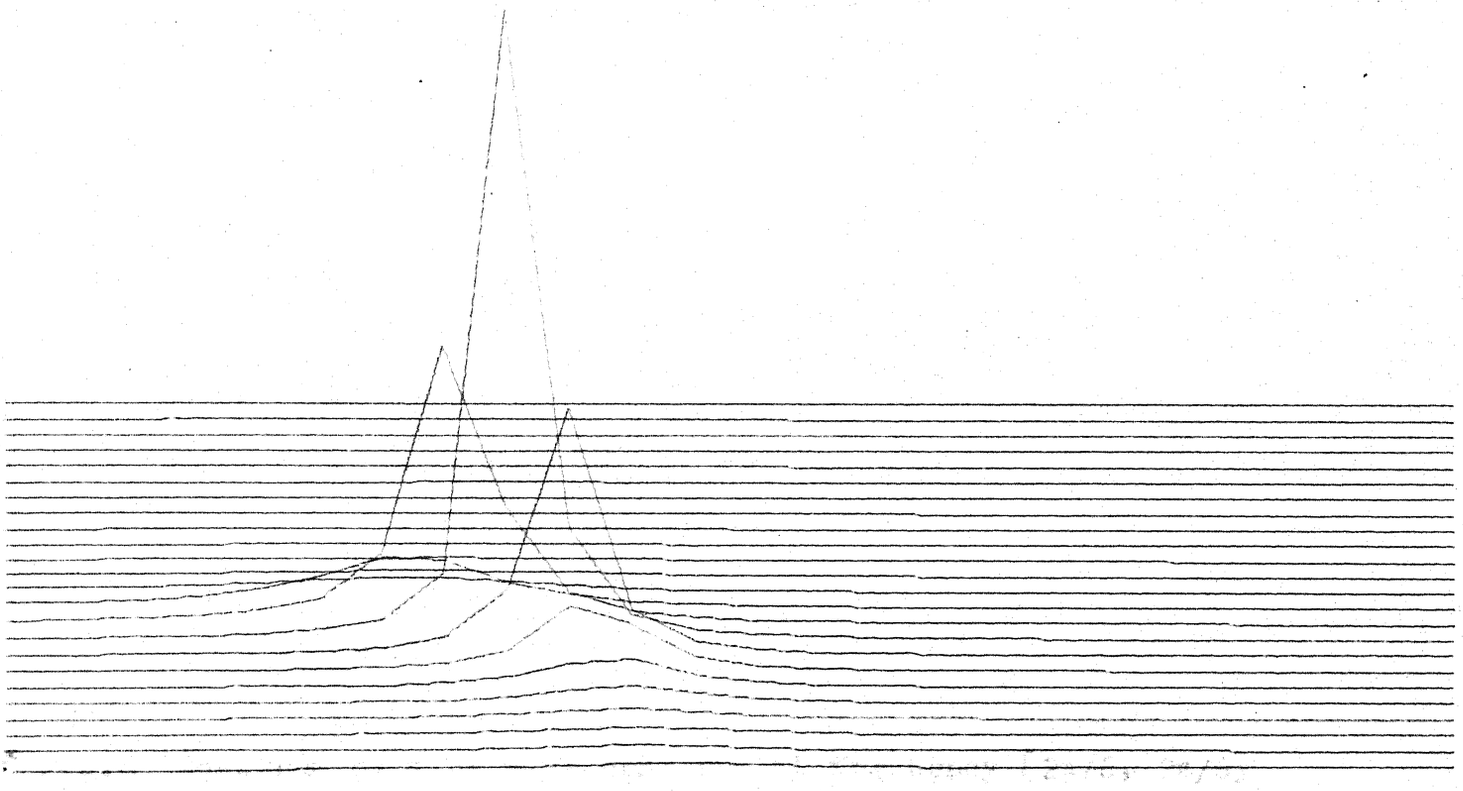

Fig.7 Spectrum of sine wave centered at frequency [2π/6, 2π/6] using AR-MEM model of order 2.

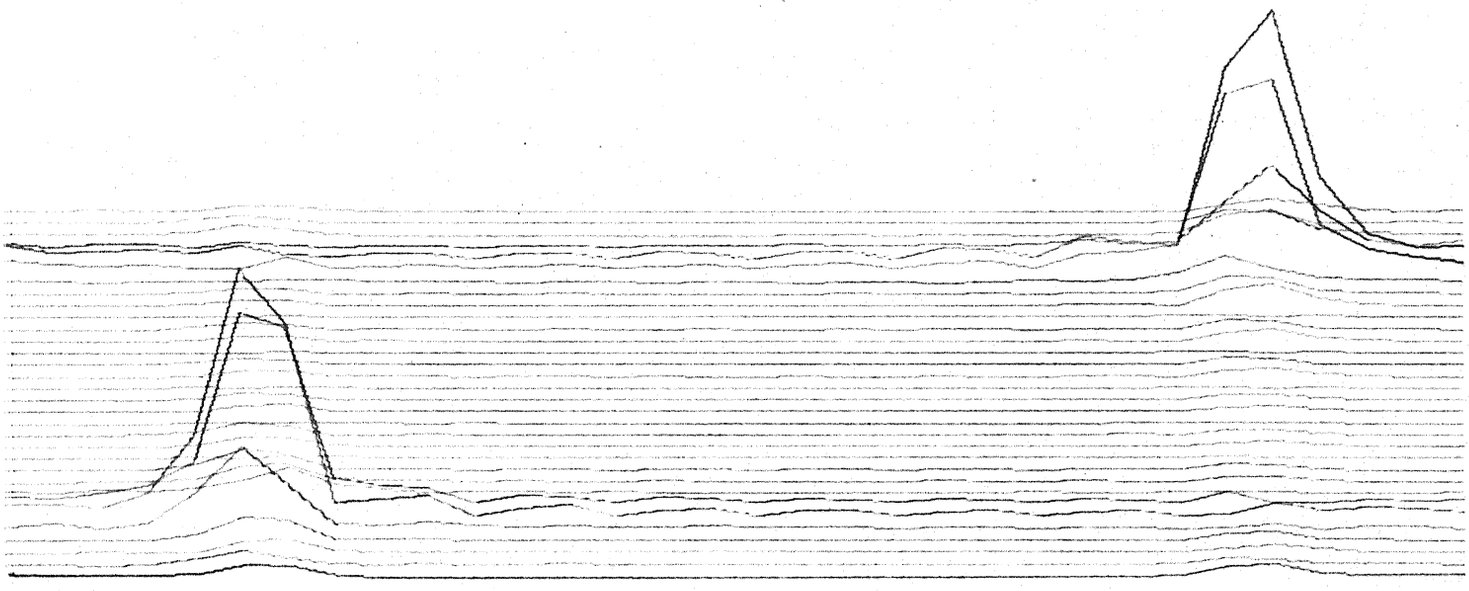

Fig.8 Spectrum of sine wave centered at frequency [2π/6, 2π/6] using B-T method with Hanning window.

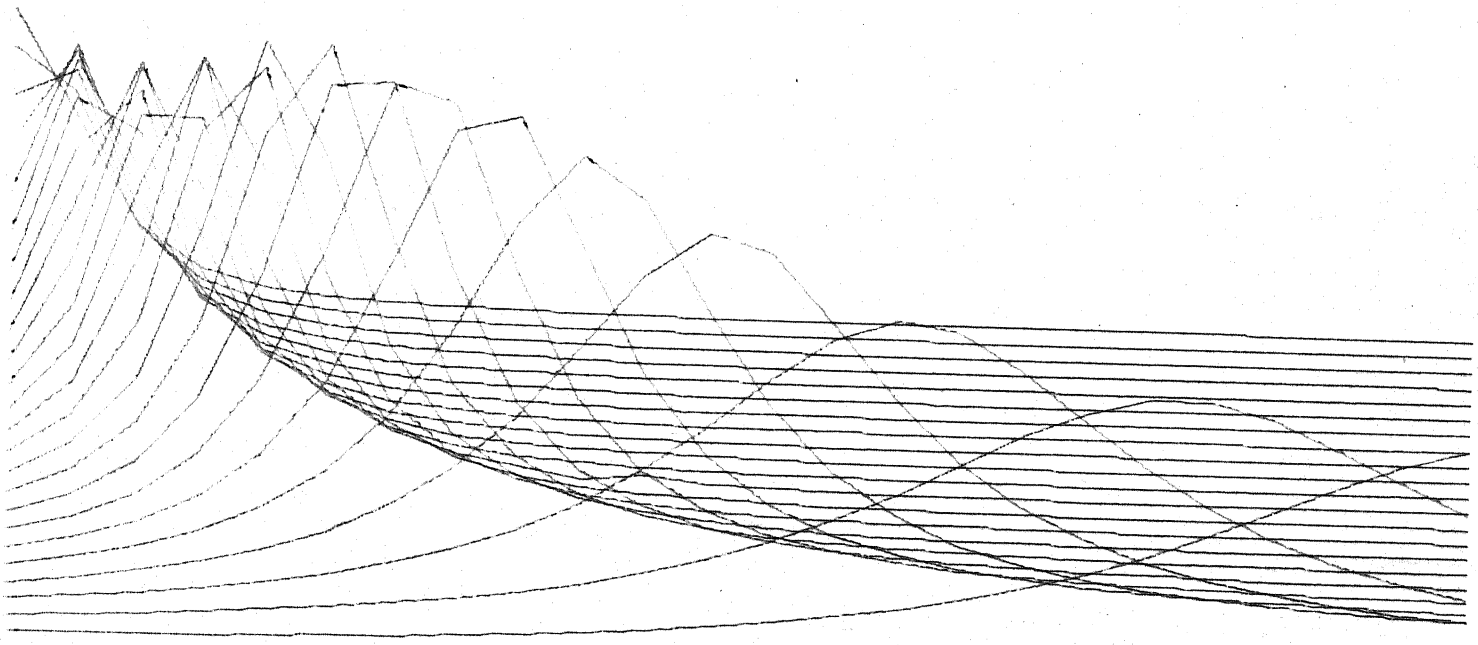

Fig.9 Resolution of composite sine wave of two frequencies $[2\pi/6, 2\pi/6]$ and $[2\pi/12, 2\pi/12]$ using AR-MEM model of order 1.

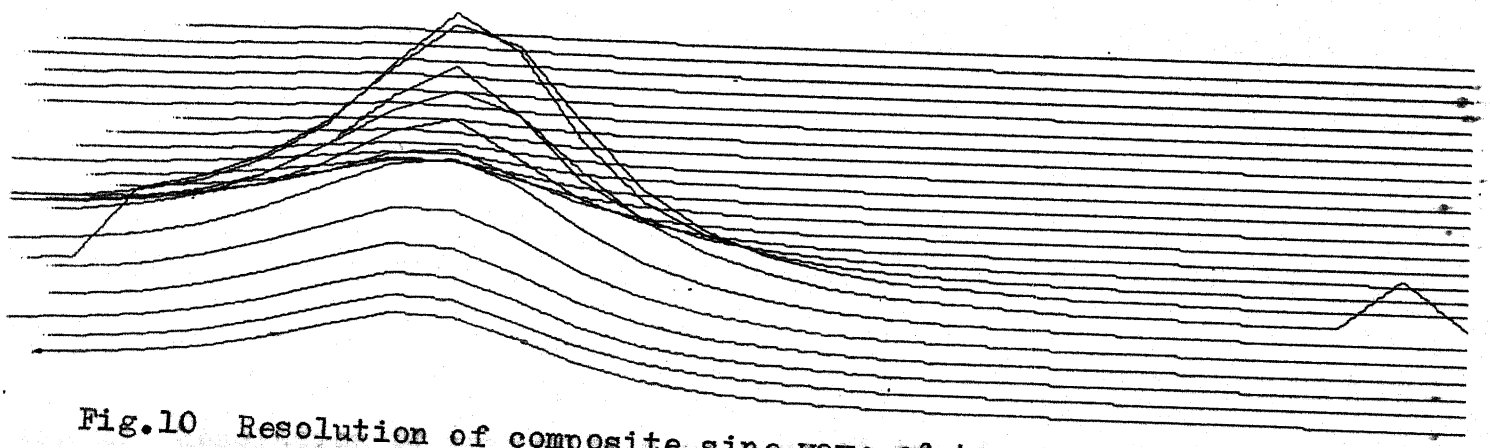

Fig.10 Resolution of composite sine wave of two frequencies $[2\pi/6, 2\pi/6]$, $[2\pi/12, 2\pi/12]$ using AR-MEM model of order 2.

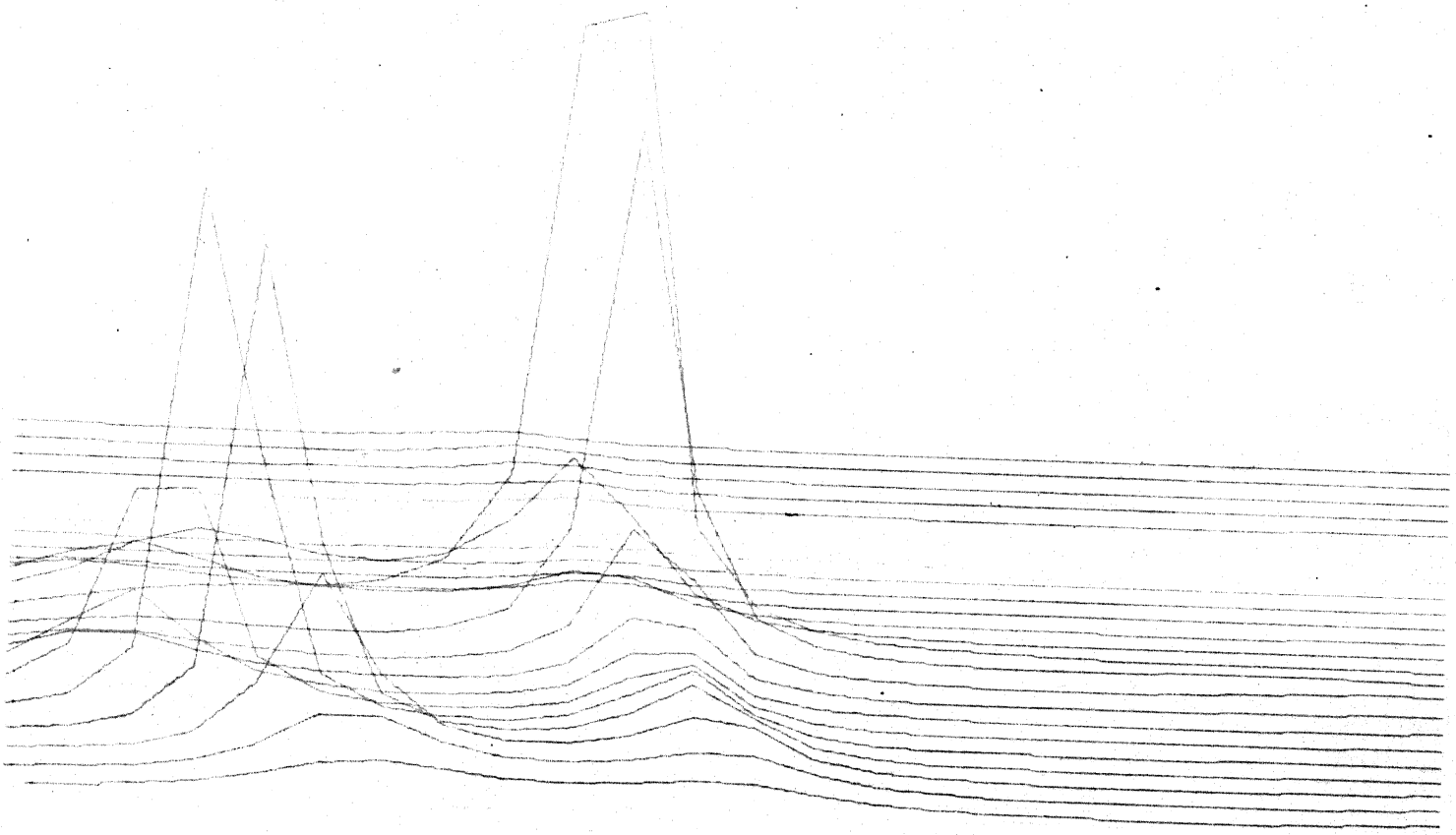

Fig.11 Resolution of composite sine wave of two frequencies $[2\pi/5, 2\pi/5]$ and $[2\pi/13, 2\pi/13]$ using AR-MEM model of order 5.

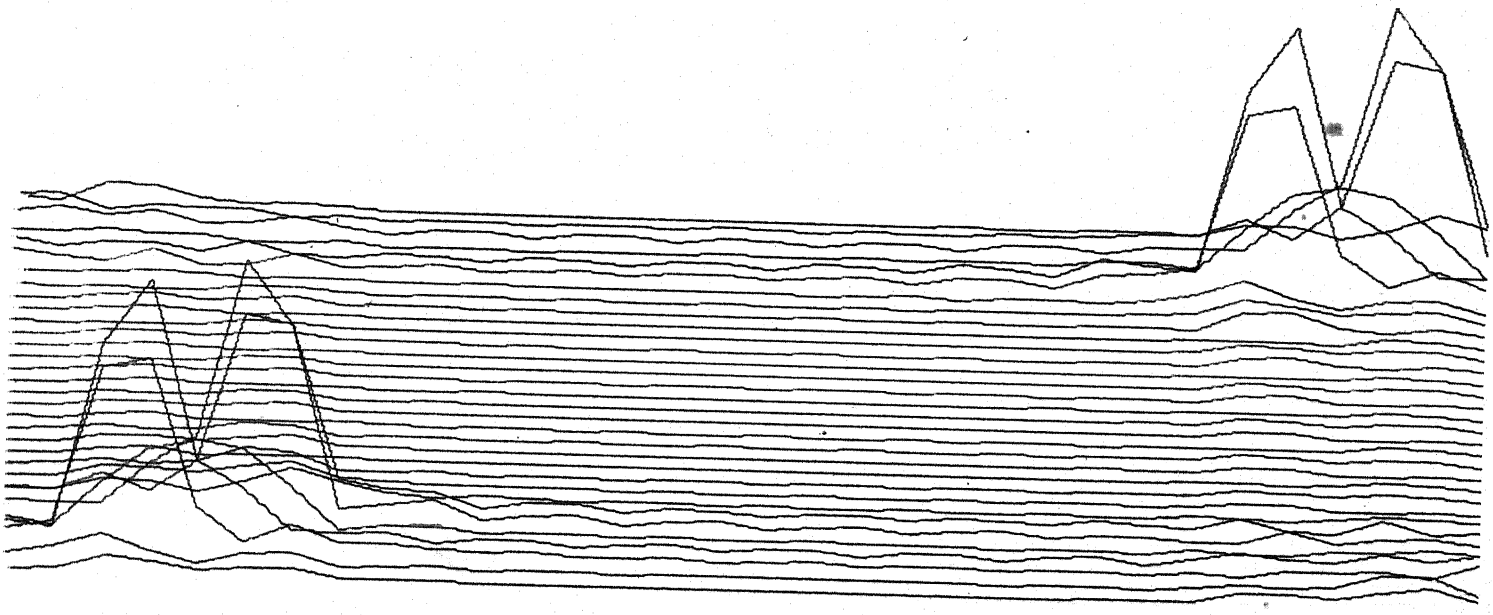

Fig.12 Resolution of composite sine wave of frequencies $[2\pi/6, 2\pi/6]$ and $[2\pi/12, 2\pi/12]$ using B-T method with Hamming window.

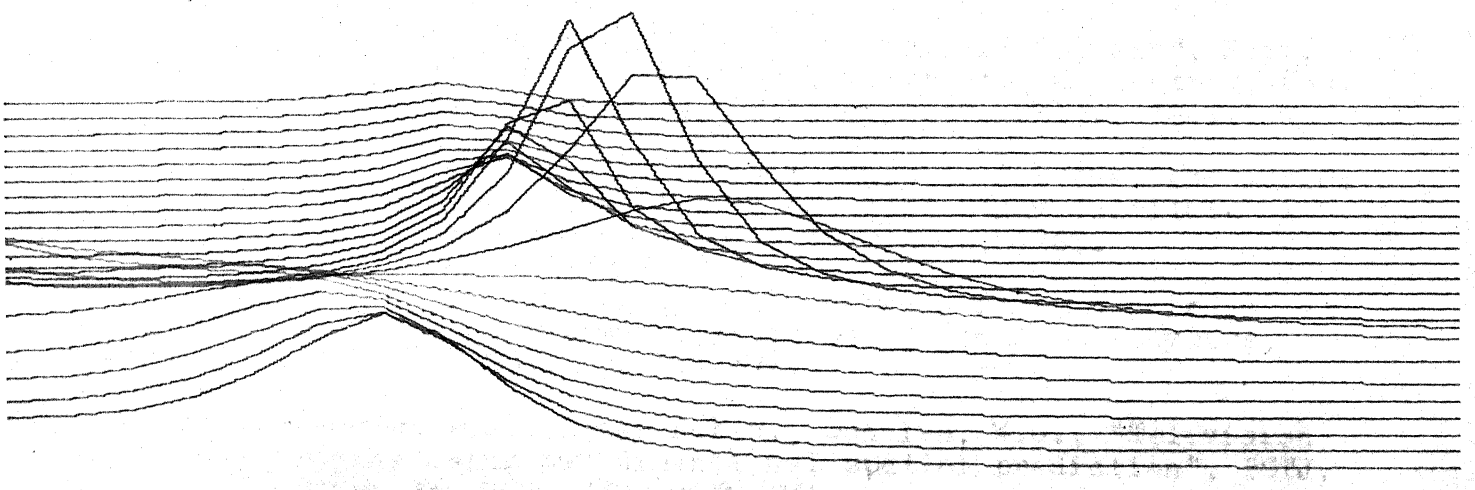

Fig.13 Resolution of composite sine wave centered at frequencies $[2\pi/5, 2\pi/5]$, $[2\pi/13, 2\pi/13]$ using AR-MEM model of order 2.

BIBLIOGRAPHY

1. Akaike, H., "Fitting Auto-Regressive models for prediction", Ann. Inst. Stat. Math., Vol.21, 1969, pp. 243-247.

2. _____, "Power spectrum estimation through autoregressive model fitting", Ann. Inst. Stat. Math., Vol.21, 1969, pp. 407-419.

3. _____, "Statistical predictor identification", Ann. Inst. Stat. Math., Vol.22, 1970, Vol.22, 1970, pp.203-217.

4. _____, "A fundamental relationship between predictor identification and power spectrum estimation", Ann.Inst. Stat.Math., Vol.22, 1970, pp. 219-223.

5. Barnard, T.E., "Analytical techniques for the computation of high-resolution wave number spectra", Adv.Anny.Res. Spec.Rep.9, Tex. Instr., Dallar, Texas, 1969, pp.IV1-IV5.

6. Blackman, R.B. and Tukey, J.W., "The Measurement of Power Spectra from the Point of view of Communication Engineering, Dover, New York, 1958.

7. Brigham, E.O;, "The Fast Fourier Transform", Prentice-Hall, New York, 1956, pp. 159-161.

8. Burg, J.P., "Maximum entropy spectral analysis", Paper presented at the 37th Annual International Meeting, Soc. of Explor. Geophysics, Oklahama city, Okla., 1967.

9. _____, "A new analysis technique for the time series data", Paper presented at Advanced Study Institute on Signal Processing, NATO, Netherlands, 1968.

10. _____, "The relationship between maximum entropy spectra and maximum likelihood spectra", Geophysics, Vol.37, 1972, pp. 375-376.

11. Connor, D.J., Pease R.F.W., Schales, W.G., "Television coding using two dimensional spatial prediction", BSTJ, Vol 50, No 1971 pp 1049-1061

12. Edward, J.A. and Fitelson,M.M., "Notes on Maximum entropy processing", Trans. IEEE Inf.Th., vol.IT-19, 1973, pp.232.

13. Huang, T.S., "Two dimensional windows", IEEE Trans. on Audio and Electroaccoustics, Vol.AU-20, pp. 80-90, March 1972.

14. Jain, A.K., "Some new techniques in image processing", Invited paper, presented at ONR symposium on Current Problems in Image Science, Naval Postgraduate School, Monterey, CA, Nov. 10-12, 1976.

15. Jaynes, E.T., "Prior probabilities", Trans. IEEE Syst. Sci., Cybes., Vol.SEC-4, 1968, pp. 227-241.

16. Jenkins, G.M. and Watts, D.G., "Spectral Analysis and its Applications, Holden Day, San Frencisco, 1968.

17. Jensen, O.G. and Ulrych, T.J., "An Analysis of the Perturbations on Barnard's Star", Astron. Jour., Vol.78, 1973, pp. 1104-1114.

18. Justice, J.H., "A Levinson's type of algorithm for two-dimensional Wiener Filtering using Biivariate Szego polynomials", Proc. IEEE, Vol.65, No.6, June 1977.

19. Koopmans, L.M., The Spectral Analysis of Time Series, Academic, New York, 1974.

20. Lacoss, R.T., "Data adaptive spectral analysis methods", Geophysics, Vol.36, 1971, pp. 661-675.

21. Levinson, N., "The Wiener RMS (Root Mean Square) Error criterion in filter design and prediction", Jour. Math. Phys., Vol.XXV, 1947, pp. 261-278.

22. Makhoul, J., "Spectral Analysis of Speech by Linear Prediction", Trans. IEEE, Elec. ACC., Vol. AU-21, 1973, pp. 140-148.

23. Newman, W.I., Extensions to the maximum entropy method, National Astronomy and Ionosphere Centre, Cornell Univ. Ithaca, 1974.

24. Newman, W.I., "A new method ofmulti-dimensions powere spectral analysis", Astronomy and Astro Phys. 54, pp. 369-380, 1977.

25. Oppenheim, A.V., and Schafer, R.W., Digital Signal Processing, Prentice-Hall, New Jersey, 1975.

26. Popuolis, A., Probability, Random Variables and Stochastic Processes, N.Y. McGraw-Hill, 1965.

27. Radar, C.M., "An improved Algorithm for high speed auto-co relation with applications to spectral estimation", Trans.IEEEE Aud. Elect., Vol.AU-18, 1970, pp. 439-441.

28. Robinson, E.A., Multichannel Time series Analysis with Digital Computer Programs, Holden Day, San Francisco, 1967.

29. Smylie, D.E., Clarke, G.K.C. and Ulrych, T.J., "Analysis of irregularities in the earth's rotation", Methods in Computational Physics, Academic, New York, Vol.13, 1973, pp. 391-430.

30. Stuller, J.A. and B. Kurz, "Two dimensional Markov representation of sampled images", IEEE Trans. on Comm. pp. 1148-1152, Oct.1976.

31. Ulrych, T.J., "Maximum entropy power spectrum of long period geomagneticreversals", Nature, Vol.235, 1972, pp. 218-219.

32. _____, "Maximum entropy power spectrum of truncated sinusoids", Jour.Geophysics Res., Vol.77, 1972, pp.1396-1400.

33. _____, Smylie, D.E., Jensen, O.G. and Clarke, G.K.C. "Predictive filtering and smoothing of short records by using Maximum entropy", Jour.Geophysics, Res., Vol.78, 1973, pp. 4959-4964.

34. _____, "and Jensen, O.G., "Cross-spectral analysis by using maximum entropy", Geophysics, Vol.39, 1974, pp. 353-354.

35. _____, and Bishop, T.N., "Maximum entropy spectral analysis and autoregressive decomposition", Reviews of Geophysics and Space Physics, Vol.13, 1975, pp. 183-200,

36. Van Den Bos, A., "Alternative interpretation of Maximum entropy spectral analysis", Trans. IEEEE Inf.Th., Vol.IT-17, 1971, pp. 493-494.

37. Vijay Kumar B.V.K., "Power Spectrum Estimation using MEM", M.Tech.Thesis, Dept.of Elect.Engg., I.I.T. Kanpur, July 1977.

38. WallaceE.L., "Statistical inference on stationary random fields"m Proc. IEEE, Vol.65, No.6, June 1977,,

39. Wiener, N., Time Series, MIT Press, Cambridge, Mass., 1964.

40. Wong, E., "Two dimensional random fieldsandrepresentation of images: SIAMJ App.Maths, Vol.16, No.4, July 1968.

41. Woods, J.W., "Two Dimensional Markov Spectral Estimation", Trans. IEEE Inf. Th., Vol.IT-22, 1976, pp. 552-559.

42. Zohar,S., "Toeplitz matrix inversion", The algorithm of W.F. French. JACM, Vol.16, No.4, Oct.1964, pp. 592-601.

43. Cooley, J.W. and Tukey, J.W., "An algorithm for machine calculation of complex Fourier series", Math Computation April 1965, Vol. 19, pp.297-301.

44. Zawalick,E.J., Fougre, P.F. and Radoski, "Spontaneous line splitting in maximum entropy power spectrum analysis", Phys. of the Earth and Planet, Int., Vol.12, 1976.

APPENDIX A

This contains the Wiener filter synthesis, based on theWiener filter theory discussed in Sec. 2.2. The algorithm used here is the one given in Section 2.3. This algorithm could not be implemented for solving the prediction equation (4.2.11) as the term $1/\gamma_{oo}$ comes on the right hand side and we cannot evaluate γ_{oo} directly. However, two examples of Wiener filter synthesis, using this algorithm are given below.

Example 1

Input array to be filtered:

$$b = \begin{bmatrix} 1 & 4 & -1 \\ 4 & 0 & 6 \end{bmatrix}$$

Desired output array:

$$c = \begin{bmatrix} 1 & -1 & -3 & 1 \\ 4 & -2 & 10 & -8 \\ 0 & 15 & 7 & 11 \\ 12 & 4 & 18 & 6 \end{bmatrix}$$

The autocorrelation of input array b:

$$\phi = \begin{bmatrix} 58 & 0 & -2 & 12 & 0 & 0 \\ 0 & 58 & 8 & -2 & 0 & 0 \\ -2 & 8 & 58 & 0 & -2 & 12 \\ 12 & -2 & 0 & 58 & 8 & -2 \\ 0 & 0 & -2 & 8 & 58 & 0 \\ 0 & 0 & 12 & -2 & 0 & 58 \end{bmatrix}$$

The cross-correlation array G is given by

$$G = \begin{bmatrix} 82 & -62 \\ 32 & 152 \\ 179 & 170 \end{bmatrix}$$

The synthesised filter array:

$$\hat{a} = \begin{bmatrix} 1.015 & -1.081 \\ 0.595 & 2.023 \\ 2.827 & 1.153 \end{bmatrix}$$

The actual filter -- 'a':

$$a = \begin{bmatrix} 1 & -1 \\ \bullet & 2 \\ 3 & 1 \end{bmatrix}$$

Example 2:

Input array:

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Desired output array:

$$c = \begin{bmatrix} 1 & 5 & 9 & 9 \\ 3 & 17 & 24 & 24 \\ -2 & 8 & 12 & 15 \\ 8 & 4 & 17 & 6 \end{bmatrix}$$

Auto-correlation matrix:

$$\phi = \begin{bmatrix} 91 & 58 & 32 & 17 & 0 & 0 \\ 58 & 91 & 83 & 32 & 0 & 0 \\ 32 & 23 & 91 & 58 & 32 & 17 \\ 17 & 32 & 58 & 91 & 23 & 32 \\ 0 & 0 & 32 & 23 & 21 & 58 \\ 0 & 0 & 17 & 32 & 58 & 91 \end{bmatrix}$$

Cross-correlation array:

$$G = \begin{bmatrix} 267 & 372 \\ 207 & 315 \\ 254 & 254 \end{bmatrix}$$

Sunthesised filter array:

$$\tilde{a} = \begin{bmatrix} 1.0 & 3.0 \\ -1.0 & 2.0 \\ 2.0 & 1.0 \end{bmatrix}$$

The actual filter:

$$a = \begin{bmatrix} 1 & 3 \\ -1 & 2 \\ 2 & 1 \end{bmatrix}$$

APPENDIX B

This contains some of the important listings of the computer subroutines used in the computer simulation. They are WHITE, COVAR, WUNIF, SPECTR, FASTFT, FFT, HAMMIN and MEM. Comment cards introduced in each routine explains the function of that routine.

```
C
C       THIS SUBROUTINE GENERATES A 2-D  WHITE NOISE ARRAY W OF SIZE
        N BY N WITH A KNOWN MEAN AND VARIANCE.
C
        DIMENSION X(625),Y(7502),W(25,25)
        RINIT=3.0
        UNIVAR=VARNS/12.
        M=(12*N)+2
        CALL WUNIF(Y,M,UNIVAR,RINIT)
        L=1
        DO 100 K=1,N
        X(K)=0.0
        LB=L+11
        DO 50 I=L,LB
50      X(K)=X(K)+Y(I)
100     L=L+12
      DO 30 I=1,25
        DO 30 J=1,25
        K=(I-1)*25+J
30      W(I,J)=X(K)
        RETURN
        END



        SUBROUTINE CCVAR(X,C,N,R)
C
CC      THIS SUBROUTINE CALCULATES THE AUTO-CORRELATION OF THE INPUT
CCC     ARRAY X OF SIZE N BY N FOR SHIFTS (-R,-R) TO (R,R).
C       THE AUTO-CORRELATION VALUES ARE STORED IN ARRAY C.
C
        DIMENSION X(24,24),C(3,3),U(24,24)
        INTEGER R
        N2=N**2
        L1=2*R+1
        DO 10 KK=1,L1
        DO 10 LL=1,L1
        K=KK-(R+1)
        L=LL-(R+1)
        C(KK,LL)=0.
        DO 15 I=1,N
        DO 15 J=1,N
        U(I,J)=F(X,I,J,K,L,N)
15      C(KK,LL)=C(KK,LL)+U(I,J)*X(I,J)
10      CONTINUE
        DO 20 I=1,L1
        DO 20 J=1,L1
20      C(I,J)=C(I,J)/FLOAT(N2)
        RETURN
        END
```

```
C
C       SUBROUTINE FOR CALCULATING THE SPECTRAL VALUES S FROM
C       THE MEM COEFFICIENTS G. UN,VN DENOTE THE NUMBER OF POINTS
C       ON THE SPECTRAL GRID. R,Q DENOTE THE ORDER OF THE AR-MEM MODEL.
C
        DIMENSION S(32,32),G(2,2)
        INTEGER R,R1,R2,Q,Q1,Q2,UN,VN
        COMPLEX SUM,Z,XX
        PI=3.1416
        R1=R+1
        Q1=Q+1
        R2=2*R+1
        Q2=2*Q+1
        DELX=1./(2.*FLOAT(UN))
        DELY=1./(2.*FLOAT(VN))
        DO 1 IUU=1,UN
        DO 1 IVV=1,VN
        IU=IUU-1
        IV=IVV-1
        SUM=(0.,0.)
        DO 2 KK=1,R1
        DO 2 LL=1,Q1
        K=KK-1
        L=LL-1
        YY=FLOAT(K)*DELX*FLOAT(IU)+FLOAT(L)*DELY*FLOAT(IV)
        Y=-2.*PI*YY
        XX=CMPLX(0.,Y)
        Z=CEXP(XX)
2       SUM=SUM+G(KK,LL)*Z
1       S(IUU,IVV)=(DELX*DELY)/(CABS(SUM)**2)
        RETURN
        END
+
        SUBROUTINE MEM(A,R,Q,G)
C
C       SUBROUTINE FOR CALCULATING THE COEFFICIENTS OF THE AR-MEM MODEL.
C       A IS THE INPUT AUTO-CORRELATION MATRIX. R AND Q DENOTE THE ORDER
C       OF THE MODEL. G IS THE OUTPUT ARRAY.
C
        DIMENSION A(20,2,2),AINV(2,2),G(3,2),C(20,2,2),D(20,2,2),B(2,1),
        DUM1(2,2),DUM2(2,2),DUM3(2,2),DUM4(2,2)
        DIMENSION OUT(2)
        INTEGER Q,Q1,Q2,Q3,R,R1,P,P1,P2,DELTA(2),P3
        R1=R+1
        Q1=Q+1
        Q2=2*Q+1
        Q3=Q-1
        N=R1
        M=N
        DO 5 I=1,Q3
        DO 5 J=1,I
```

```
      P=I*Q+J
      DO 4 K=1,R1
      DO 4 L=1,R1
      D(P,K,L)=0.
4     C(P,K,L)=0.
5     CONTINUE
      DELTA(1)=1
      DO 20 I=2,R1
20    DELTA(I)=0
      DO 30 I=1,R1
      DO 30 J=1,R1
30    AINV(I,J)=A(Q1,I,J)
      CALL MATINV(AINV,N,B,O,DETERM)
      P=1
      DO 40 I=1,R1
      DO 40 J=1,R1
      K=I+Q+1
40    DUM1(I,J)=A(K,I,J)
      DO 45 I=1,R1
      DO 45 J=1,R1
45    DUM2(I,J)=AINV(I,J)
      CALL MULTI(DUM2,DUM1,N,M)
      DO 50 I=1,R1
      DO 50 J=1,R1
50    C(P,I,J)=DUM2(I,J)
      DO 60 I=1,R1
      DO 60 J=1,R1
      K=Q
60    DUM1(I,J)=A(K,I,J)
      DO 65 I=1,R1
      DO 65 J=1,R1
65    DUM2(I,J)=AINV(I,J)
      CALL MULTI(DUM2,DUM1,N,M)
      DO 70 I=1,R1
      DO 70 J=1,R1
70    D(P,I,J)=DUM2(I,J)
      IF(Q.EQ.1) GO TO 77
      DO 80 NN1=1,Q3
      N1=NN1-1
      J1=NN1+1
      DO 85 I=1,R1
      DO 85 J=1,R1
85    DUM3(I,J)=0.
      DO 90 NN2=1,J1
      N2=NN2-1
      P=N2*Q+NN1
      K=NN1-N2+Q1
      DO 100 I=1,R1
      DO 100 J=1,R1
      DUM1(I,J)=A(K,I,J)
100   DUM2(I,J)=D(P,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      DO 95 I=1,R1
      DO 95 J=1,R1
95    DUM3(I,J)=DUM3(I,J)+DUM1(I,J)
90    CONTINUE
```

```
      K=Q1
      DO 110 I=1,R1
      DO 110 J=1,R1
110   DUM1(I,J)=A(K,I,J)
      L=0
      CALL MATADD(DUM3,DUM1,L,N)
      CALL MATINV(DUM3,N,B,0,DETERM)
      DO 115 I=1,R1
      DO 115 J=1,R1
115   DUM4(I,J)=0.
      DO 120 NN3=2,NN1
      N3=NN3-1
      K=NN3+Q1
      P=N3*Q+NN1
      DO 130 I=1,R1
      DO 130 J=1,R1
      DUM1(I,J)=A(K,I,J)
130   DUM2(I,J)=C(P,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      DO 125 I=1,R1
      DO 125 J=1,R1
125   DUM4(I,J)=DUM4(I,J)+DUM1(I,J)
120   CONTINUE
      K=Q1+NN1+1
      DO 140 I=1,R1
      DO 140 J=1,R1
140   DUM1(I,J)=A(K,I,J)
      CALL MATADD(DUM4,DUM1,L,N)
      CALL MULTI(DUM3,DUM4,N,M)
      P=NN1+1
      DO 150 I=1,R1
      DO 150 J=1,R1
150   C(P,I,J)=DUM3(I,J)
      DO 155 I=1,R1
      DO 155 J=1,R1
155   DUM3(I,J)=0.
      DO 160 NN2=1,J1
      N2=NN2-1
      P=N2*Q+NN1
      K=N2-NN1+Q1
      DO 170 I=1,R1
      DO 170 J=1,R1
      DUM1(I,J)=A(K,I,J)
170   DUM2(I,J)=C(P,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      DO 175 I=1,R1
      DO 175 J=1,R1
175   DUM3(I,J)=DUM3(I,J)+DUM1(I,J)
160   CONTINUE
      K=Q1
      DO 180 I=1,R1
      DO 180 J=1,R1
180   DUM1(I,J)=A(K,I,J)
      CALL MATADD(DUM3,DUM1,L,N)
      CALL MATINV(DUM3,N,B,0,DETERM)
      DO 185 I=1,R1
```

```
      DO 185 J=1,R1
185   DUM4(I,J)=0.
      DO 190 NN3=1,N-1
      N3=NN3-1
      K=Q1-NN3
      P=N3*Q+NN1
      DO 200 I=1,R1
      DO 200 J=1,R1
      DUM1(I,J)=A(K,I,J)
200   DUM2(I,J)=D(P,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      DO 205 I=1,R1
      DO 205 J=1,R1
205   DUM4(I,J)=DUM4(I,J)+DUM1(I,J)
190   CONTINUE
      K=Q1-(NN1+1)
      DO 210 I=1,R1
      DO 210 J=1,R1
210   DUM1(I,J)=A(K,I,J)
      CALL MATADD(DUM4,DUM1,L,N)
      CALL MULTI(DUM3,DUM4,N,M)
      P=NN1+1
      DO 220 I=1,R1
      DO 220 J=1,R1
220   D(P,I,J)=DUM3(I,J)
      DO 230 NN2=1,NN1
      N2=NN2-1
      P=NN2*Q+NN1+1
      P1=N2*Q+NN1
      P2=(N1-N2)*Q+NN1
      P3=NN1+1
      DO 240 I=1,R1
      DO 240 J=1,R1
      DUM1(I,J)=D(P2,I,J)
      DUM2(I,J)=C(P3,I,J)
240   DUM3(I,J)=C(P1,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      CALL MATADD(DUM3,DUM1,L,N)
      DO 250 I=1,R1
      DO 250 J=1,R1
250   C(P,I,J)=DUM3(I,J)
      DO 260 I=1,R1
      DO 260 J=1,R1
      DUM1(I,J)=C(P2,I,J)
      DUM2(I,J)=D(P3,I,J)
260   DUM4(I,J)=D(P1,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      CALL MATADD(DUM4,DUM1,L,N)
      DO 270 I=1,R1
      DO 270 J=1,R1
270   D(P,I,J)=DUM4(I,J)
230   CONTINUE
80    CONTINUE
77    CONTINUE
      P=Q*Q+Q
      DO 245 I=1,R1
```

```
      DO 245 J=1,R1
      IF(I.EQ.J) GO TO 255
      D(P,I,J)=0.
      GO TO 245
255   D(P,I,J)=-1
245   CONTINUE
      DO 265 I=1,R1
      DO 265 J=1,R1
265   DUM3(I,J)=0.
      DO 275 NN1=1,Q
      N1=NN1-1
      P=N1*Q+Q
      K=Q-N1+Q1
      DO 280 I=1,R1
      DO 280 J=1,R1
      DUM1(I,J)=A(K,I,J)
280   DUM2(I,J)=D(P,I,J)
      CALL MULTI(DUM1,DUM2,N,M)
      DO 290 I=1,R1
      DO 290 J=1,R1
290   DUM3(I,J)=DUM3(I,J)+DUM1(I,J)
275   CONTINUE
      DO 300 I=1,R1
      DO 300 J=1,R1
      K=Q+1
300   DUM1(I,J)=A(K,I,J)
      CALL MATADD(DUM1,DUM3,L,N)
      CALL MATINV(DUM1,N,B,0,DETERM)
      DO 310 II=1,Q1
      I=II-1
      P=(Q-1)*Q+Q
      DO 320 K=1,R1
      DO 320 L=1,R1
320   DUM2(K,L)=D(P,K,L)
      CALL MULTI(DUM2,DUM1,N,M)
      CALL VECTOR(DUM2,DELTA,OUT,N,M)
      DO 330 K=1,R1
330   G(II,K)=-OUT(K)
310   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULTI(A,B,N,M)
      DIMENSION A(2,2),B(2,2),C(2,2)
      DO 10 I=1,N
      DO 10 J=1,N
      C(I,J)=0.
      DO 10 K=1,N
10    C(I,J)=C(I,J)+A(I,K)*B(K,J)
      DO 20 I=1,N
      DO 20 J=1,N
20    A(I,J)=C(I,J)
```

```
      RETURN
      END



      SUBROUTINE MATADD(A,B,L,N)
      DIMENSION A(2,2),B(2,2)
      DO 10 I=1,N
      DO 10 J=1,N
10    A(I,J)=A(I,J)-B(I,J)
      RETURN
      END



      SUBROUTINE VECTOR(A,B,OUT,N,M)
      DIMENSION A(2,2),OUT(2)
      INTEGER B(2)
      DO 10 I=1,N
      OUT(I)=0.
      DO 10 J=1,N
10    OUT(I)=OUT(I)+A(I,J)*(FLOAT(B(J)))
      RETURN
      END



      SUBROUTINE HAMMIN(R,N)
      COMPLEX R(N,N)
      P=32.*32.*2.
      ANN=SQRT(P)
      PI=3.14159
      DO 10 II=1,N
      DO 10 JJ=1,N
      I=II-1
      J=JJ-1
      K=I**2+J**2
      XX=FLOAT(K)
      X=SQRT(XX)
      THETA=(2.*PI/ANN)*X
      W1=0.54-0.46*COS(THETA)
10    R(II,JJ)=R(II,JJ)*W1
      RETURN
      END



      SUBROUTINE FASTFT(A,N,NW)
```

```
C
C       THIS SUBROUTINE CALCULATES THE 2-D FFT OF THE INPUT ARRAY A.
C
        COMPLEX A(NW,NW),P(32),Q(32),TEMP(32)
        M=5
        DO 200 I=1,N
        DO 201 K=1,N
201     TEMP(K)=A(I,K)
        CALL FFT(TEMP,N,M)
        DO 202 K=1,N
202     A(I,K)=TEMP(K)
200     CONTINUE
        DO 300 J=1,N
        DO 301 K=1,N
301     TEMP(K)=A(K,J)
        CALL FFT(TEMP,N,M)
        DO 302 K=1,N
302     A(K,J)=TEMP(K)
300     CONTINUE
        RETURN
        END
```

```
        SUBROUTINE FFT(A,N,NU)
C
C       THIS SUBROUTINE COMPUTES THE FFT OF A SEQUENCE A OF LENGTH N.
        THE LENGTH N SHOULD BE A POWER OF 2. NU IS LOGARITHM OF N TO BASE
C
        COMPLEX A(N)
        DIMENSION XREAL(100),XIMAG(100)
        DO 10 I=1,N
        XIMAG(I)=AIMAG(A(I))
        XREAL(I)=REAL(A(I))
10      CONTINUE
      N2=N/2
      NU1=NU-1
      K=0
      DO100L=1,NU
102   DO101 I=1,N2
      P=IBITR(K/2**NU1,NU)
      ARG=6.283185*P/FLOAT(N)
      C=COS(ARG)
      S=SIN(ARG)
      K1=K+1
      K1N2=K1+N2
      TREAL=XREAL(K1N2)*C+XIMAG(K1N2)*S
      TIMAG=XIMAG(K1N2)*C-XREAL(K1N2)*S
      XREAL(K1N2)=XREAL(K1)-TREAL
      XIMAG(K1N2)=XIMAG(K1)-TIMAG
      XREAL(K1)=XREAL(K1)+TREAL
      XIMAG(K1)=XIMAG(K1)+TIMAG
101   K=K+1
      K=K+N2
```

```
      IF(K.LT.N)   GOTO102
      K=0
      NU1=NU1-1
100   N2=N2/2
      DO103K=1,N
      I=IBITR(K-1,NU)+1
      IF(I.LE.K)   GOTO103
      TREAL=XREAL(K)
      TIMAG=XIMAG(K)
      XREAL(K)=XREAL(I)
      XIMAG(K)=XIMAG(I)
      XREAL(I)=TREAL
      XIMAG(I)=TIMAG
103   CONTINUE
        DO 104 I=1,N
        RE=XREAL(I)
        AI=XIMAG(I)
104     A(I)=CMPLX(RE,AI)
      RETURN
      END
```

```
      FUNCTION  IBITR(J,NU)
      J1=J
      IBITR=0
      DO200I=1,NU
      J2=J1/2
      IBITR=IBITR*2+(J1-2*J2)
200   J1=J2
      RETURN
      END
```